A gentle introduction to quantum complexity theory[1]

Sevag Gharibian

Department of Computer Science
Institute for Photonic Quantum Systems (PhoQS)
Paderborn University
Germany

[1] https://groups.uni-paderborn.de/fg-qi/courses/UPB_QCOMPLEXITY/2020/UPB_QCOMPLEXITY_syllabus.html for course notes/Youtube videos

Goal

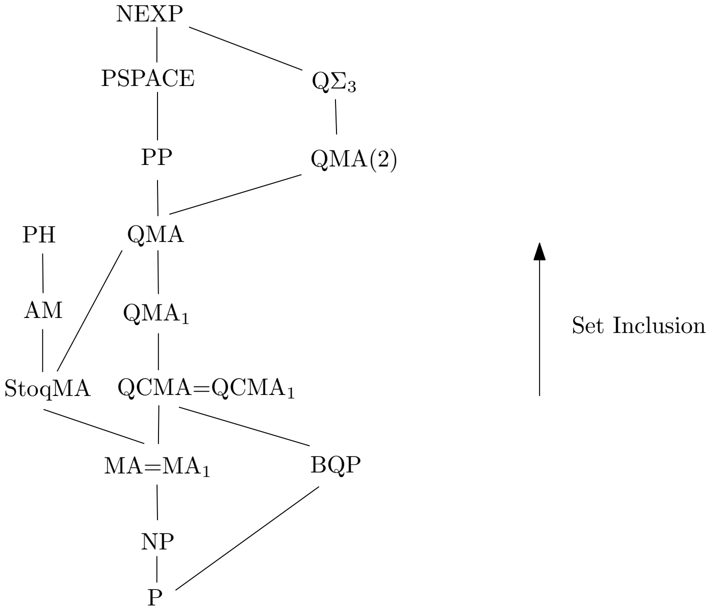
Goal

Goal

High-level steps:

1. Formal model for computation
2. Complexity theory within this model (classical and quantum)

Preview


NEXP
PSPACE
QΣ3
PP
QMA(2)
PH
QMA
AM
QMA1
StoqMA
QCMA=QCMA1
MA=MA1
BQP
NP
P
Set Inclusion

Outline

Outline

Computational complexity theory

What resources (e.g. time, space, communication, etc) are required to solve a given computational problem?

Computational complexity theory

What resources (e.g. time, space, communication, etc) are required to solve a given computational problem?

Conversation with first-year CS undergrad:

- Instructor: What does it mean to compute shortest path from point A to point B on a map in $O(n^2)$ time?

Computational complexity theory

What resources (e.g. time, space, communication, etc) are required to solve a given computational problem?

Conversation with first-year CS undergrad:

- Instructor: What does it mean to compute shortest path from point A to point B on a map in $O(n^2)$ time?
- Student: Given a map, A and B, output the shortest path using approximately n^2 steps.

Computational complexity theory

What resources (e.g. time, space, communication, etc) are required to solve a given computational problem?

Conversation with first-year CS undergrad:

- Instructor: What does it mean to compute shortest path from point A to point B on a map in $O(n^2)$ time?
- Student: Given a map, A and B, output the shortest path using approximately n^2 steps.
- Instructor: What do you mean by "steps"?

Computational complexity theory

What resources (e.g. time, space, communication, etc) are required to solve a given computational problem?

Conversation with first-year CS undergrad:

- Instructor: What does it mean to compute shortest path from point A to point B on a map in $O(n^2)$ time?
- Student: Given a map, A and B, output the shortest path using approximately n^2 steps.
- Instructor: What do you mean by “steps”?
- Student: You know, loop iterations... in Java or something.

Computational complexity theory

What resources (e.g. time, space, communication, etc) are required to solve a given computational problem?

Conversation with first-year CS undergrad:

- Instructor: What does it mean to compute shortest path from point A to point B on a map in $O(n^2)$ time?
- Student: Given a map, A and B, output the shortest path using approximately n^2 steps.
- Instructor: What do you mean by “steps”?
- Student: You know, loop iterations... in Java or something.
- Instructor: What if I don’t code in Java? What if I code in C, or machine code?

Computational complexity theory

What resources (e.g. time, space, communication, etc) are required to solve a given computational problem?

Conversation with first-year CS undergrad:

- Instructor: What does it mean to compute shortest path from point A to point B on a map in $O(n^2)$ time?
- Student: Given a map, A and B, output the shortest path using approximately n^2 steps.
- Instructor: What do you mean by "steps"?
- Student: You know, loop iterations... in Java or something.
- Instructor: What if I don't code in Java? What if I code in C, or machine code?
- Student: Ummmmm...

Computational complexity theory

What resources (e.g. time, space, communication, etc) are required to solve a given computational problem?

Conversation with first-year CS undergrad:

- Instructor: What does it mean to compute shortest path from point A to point B on a map in $O(n^2)$ time?
- Student: Given a map, A and B, output the shortest path using approximately n^2 steps.
- Instructor: What do you mean by “steps”?
- Student: You know, loop iterations... in Java or something.
- Instructor: What if I don’t code in Java? What if I code in C, or machine code?
- Student: Ummmmm...
- Instructor: Do you even know what n is?

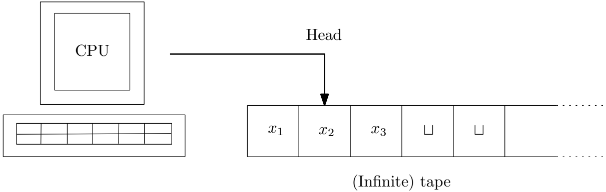
Computational complexity theory

What resources (e.g. time, space, communication, etc) are required to solve a given computational problem?

Conversation with first-year CS undergrad:

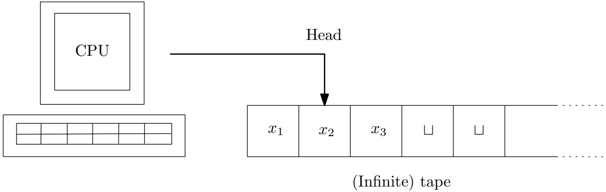
- Instructor: What does it mean to compute shortest path from point A to point B on a map in $O(n^2)$ time?
- Student: Given a map, A and B, output the shortest path using approximately n^2 steps.
- Instructor: What do you mean by “steps”?
- Student: You know, loop iterations... in Java or something.
- Instructor: What if I don’t code in Java? What if I code in C, or machine code?
- Student: Ummmmm...
- Instructor: Do you even know what n is?
- Student:

Turing machine (TM)



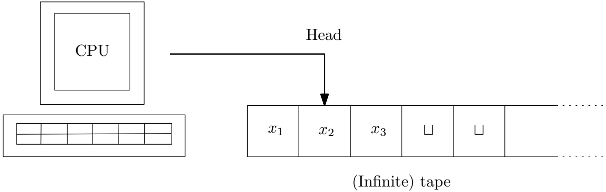
- Before TM starts: Input $x \in \{0,1\}^*$ written on tape ($\sqcup$ are blank cells)

Turing machine (TM)



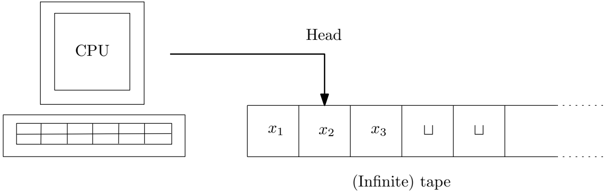
- Before TM starts: Input $x \in \{0,1\}^*$ written on tape ($\sqcup$ are blank cells)
- One step of computation:
  - Head at position i reads bit $b_i \in \{0,1\}^*$ on tape
  - Write bit $b_i' \in \{0,1\}$ to position i on tape
  - Move head left or right 1 cell

Turing machine (TM)



- Before TM starts: Input $x \in \{0,1\}^*$ written on tape ($\sqcup$ are blank cells)
- One step of computation:
  - Head at position i reads bit $b_i \in \{0,1\}^*$ on tape
  - Write bit $b_i' \in \{0,1\}$ to position i on tape
  - Move head left or right 1 cell
- What does it mean to "compute"? Given input x, TM runs for some number of steps, and either:
  - Halts $\rightarrow$ tape contents are "output" of computation
  - Doesn't halt $\rightarrow$ infinite loop

Turing machine (TM)



- Before TM starts: Input $x \in \{0,1\}^*$ written on tape ($\sqcup$ are blank cells)
- One step of computation:
  - Head at position i reads bit $b_i \in \{0,1\}^*$ on tape
  - Write bit $b_i' \in \{0,1\}$ to position i on tape
  - Move head left or right 1 cell
- What does it mean to "compute"? Given input x, TM runs for some number of steps, and either:
  - Halts $\rightarrow$ tape contents are "output" of computation
  - Doesn't halt $\rightarrow$ infinite loop
- What is n? Input size, i.e. $x \in \{0,1\}^n$.

Why Turing machines (TMs)?

Why Turing machines (TMs)?

- Simple model to state and understand:
  - Computation time: Number of steps for TM to halt on input x
  - Computation space: Number of tape cells TM uses on tape

Why Turing machines (TMs)?

- Simple model to state and understand:
  - Computation time: Number of steps for TM to halt on input x
  - Computation space: Number of tape cells TM uses on tape
- Robust: Power of model unchanged under minor modifications (e.g. 2 tapes instead of 1)

Why Turing machines (TMs)?

- Simple model to state and understand:
  - Computation time: Number of steps for TM to halt on input x
  - Computation space: Number of tape cells TM uses on tape
- Robust: Power of model unchanged under minor modifications (e.g. 2 tapes instead of 1)
- Church-Turing thesis:

  If there exists a mechanical process for computing function $f : \{0,1\}^* \to \{0,1\}^*$, then there exists a Turing machine computing f.

Outline

Decision problems

Informally: “Problems with a YES or NO answer.”

Decision problems

Informally: "Problems with a YES or NO answer."

Decision problem $A = (A_{\text{yes}}, A_{\text{no}})$

Suppose $A_{\text{yes}} \cup A_{\text{no}}$ partition $\{0,1\}^*$, i.e. A_{yes} are "YES" instances, A_{no} the "NO" instances.

Given input $x \in \{0,1\}^*$,

- if $x \in A_{\text{yes}}$, output YES.
- if $x \in A_{\text{no}}$, output NO.

Decision problems

Informally: "Problems with a YES or NO answer."

Decision problem $A = (A_{\text{yes}}, A_{\text{no}})$

Suppose $A_{\text{yes}} \cup A_{\text{no}}$ partition $\{0,1\}^*$, i.e. A_{yes} are "YES" instances, A_{no} the "NO" instances.

Given input $x \in \{0,1\}^*$,

- if $x \in A_{\text{yes}}$, output YES.
- if $x \in A_{\text{no}}$, output NO.

Example: Integer multiplication (MULTIPLY).

- Non-decision problem formulation: Given $(x, y) \in \mathbb{Z}^2$, what is xy?

Decision problems

Informally: "Problems with a YES or NO answer."

Decision problem $A = (A_{\text{yes}}, A_{\text{no}})$

Suppose $A_{\text{yes}} \cup A_{\text{no}}$ partition $\{0,1\}^*$, i.e. A_{yes} are "YES" instances, A_{no} the "NO" instances.

Given input $x \in \{0,1\}^*$,

- if $x \in A_{\text{yes}}$, output YES.
- if $x \in A_{\text{no}}$, output NO.

Example: Integer multiplication (MULTIPLY).

- Non-decision problem formulation: Given $(x, y) \in \mathbb{Z}^2$, what is xy?
- Decision problem formulation: Given $(x, y, t) \in \mathbb{Z}^3$, is $xy \leq t$?

Decision problems

Informally: "Problems with a YES or NO answer."

Decision problem $A = (A_{\text{yes}}, A_{\text{no}})$

Suppose $A_{\text{yes}} \cup A_{\text{no}}$ partition $\{0,1\}^*$, i.e. A_{yes} are "YES" instances, A_{no} the "NO" instances.

Given input $x \in \{0,1\}^*$,

- if $x \in A_{\text{yes}}$, output YES.
- if $x \in A_{\text{no}}$, output NO.

Example: Integer multiplication (MULTIPLY).

- Non-decision problem formulation: Given $(x, y) \in \mathbb{Z}^2$, what is xy?
- Decision problem formulation: Given $(x, y, t) \in \mathbb{Z}^3$, is $xy \leq t$?
- Formally,

$$A_{\text{yes}} = \left\{(x, y, t) \in \mathbb{Z}^3 \mid xy \leq t\right\}, \qquad A_{\text{no}} = \{0,1\}^* \setminus A_{\text{yes}}.$$

Polynomial-Time (P)

Polynomial-Time (P)

Decision problem $A = (A_{\text{yes}}, A_{\text{no}})$ is in P if there exists TM M and polynomial p, such that for any input $x \in \{0,1\}^n$, M halts in a most $O(p(n))$ steps and:

- (YES case) If $x \in A_{\text{yes}}$, M outputs 1, i.e. $M(x) = 1$.
- (NO case) If $x \in A_{\text{no}}$, M outputs 0, i.e. $M(x) = 0$.

Polynomial-Time (P)

Polynomial-Time (P)

Decision problem $A = (A_{\text{yes}}, A_{\text{no}})$ is in P if there exists TM M and polynomial p, such that for any input $x \in \{0,1\}^n$, M halts in a most $O(p(n))$ steps and:

- (YES case) If $x \in A_{\text{yes}}$, M outputs 1, i.e. $M(x) = 1$.
- (NO case) If $x \in A_{\text{no}}$, M outputs 0, i.e. $M(x) = 0$.

MULTIPLY: $A_{\text{yes}} = \{(x, y, t) \in \mathbb{Z}^3 \mid xy \leq t\}$.

- Grade-school multiplication algorithm on TM takes $O(n^2)$ steps $\Rightarrow$ MULTIPLY $\in$ P.
- Aside: Fastest multiplication algorithm is $O(n \log n)$ [Harvey, van der Hoeven, 2019].

First cousins

MULTIPLY: $A_{\text{yes}} = \{(x, y, t) \in \mathbb{Z}^3 \mid xy \leq t\}$.

- Grade-school multiplication algorithm on TM takes $O(n^2)$ steps $\Rightarrow$ MULTIPLY $\in$ P.
- Aside: Fastest multiplication algorithm is $O(n \log n)$ [Harvey, van der Hoeven, 2019].

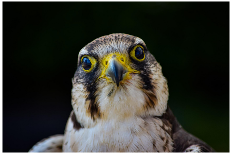
First cousins

MULTIPLY: $A_{\text{yes}} = \{(x, y, t) \in \mathbb{Z}^3 \mid xy \leq t\}$.

- Grade-school multiplication algorithm on TM takes $O(n^2)$ steps $\Rightarrow$ MULTIPLY $\in$ P.
- Aside: Fastest multiplication algorithm is $O(n \log n)$ [Harvey, van der Hoeven, 2019].

FACTOR: $A_{\text{yes}} = \{(x, t) \in \mathbb{Z}^2 \mid x \geq 0 \text{ has non-trivial factor } \leq t\}$.

- Is FACTOR $\in$ P?



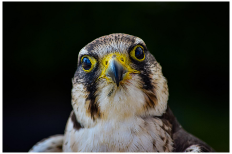
First cousins

MULTIPLY: $A_{\text{yes}} = \{(x, y, t) \in \mathbb{Z}^3 \mid xy \leq t\}$.

- Grade-school multiplication algorithm on TM takes $O(n^2)$ steps $\Rightarrow$ MULTIPLY $\in$ P.
- Aside: Fastest multiplication algorithm is $O(n \log n)$ [Harvey, van der Hoeven, 2019].

FACTOR: $A_{\text{yes}} = \{(x, t) \in \mathbb{Z}^2 \mid x \geq 0 \text{ has non-trivial factor } \leq t\}$.

- Is FACTOR $\in$ P?



- Strongly believed FACTOR $\notin$ P (security of popular cryptosystem RSA relies on it)

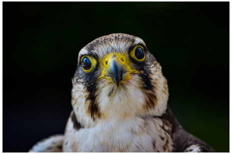
First cousins

MULTIPLY: $A_{\text{yes}} = \{(x, y, t) \in \mathbb{Z}^3 \mid xy \leq t\}$.

- Grade-school multiplication algorithm on TM takes $O(n^2)$ steps $\Rightarrow$ MULTIPLY $\in$ P.
- Aside: Fastest multiplication algorithm is $O(n \log n)$ [Harvey, van der Hoeven, 2019].

FACTOR: $A_{\text{yes}} = \{(x, t) \in \mathbb{Z}^2 \mid x \geq 0$ has non-trivial factor $\leq t\}$.

- Is FACTOR $\in$ P?



- Strongly believed FACTOR $\notin$ P (security of popular cryptosystem RSA relies on it)
- Can be verified easily: Given claimed "proof" $y \in \mathbb{Z}$, can efficiently check if $y \leq t$ and $x \mod y = 0$.

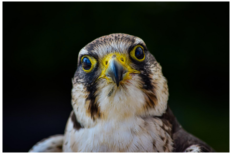
First cousins

MULTIPLY: $A_{\text{yes}} = \{(x, y, t) \in \mathbb{Z}^3 \mid xy \leq t\}$.

- Grade-school multiplication algorithm on TM takes $O(n^2)$ steps $\Rightarrow$ MULTIPLY $\in$ P.
- Aside: Fastest multiplication algorithm is $O(n \log n)$ [Harvey, van der Hoeven, 2019].

FACTOR: $A_{\text{yes}} = \{(x, t) \in \mathbb{Z}^2 \mid x \geq 0 \text{ has non-trivial factor } \leq t\}$.

- Is FACTOR $\in$ P?



- Strongly believed FACTOR $\notin$ P (security of popular cryptosystem RSA relies on it)
- Can be verified easily: Given claimed "proof" $y \in \mathbb{Z}$, can efficiently check if $y \leq t$ and $x \mod y = 0$.

Bonus: In contrast to FACTOR, checking if x has *any* non-trivial factor is in P [Agrawal, Kayal, Saxena 2002]

Sanity check

Why isn't the naive brute force algorithm poly-time?

Input: $(x, t) \in \mathbb{Z}^2$
Output: Factor $y \in \mathbb{Z}$ of x with $y \leq t$, if one exists

1. Set $k = 2$
2. While $(k < t)$
   a) If $x \mod k = 0$ then return k
   b) $k = k + 1$
3. Return "no factor found"

Runtime: $\approx$ num loop iterations $O(x)$.

Exercise 1: Why is this not poly-time?

Non-deterministic Polynomial-Time (NP)

Polynomial-Time (P)

Decision problem $A = (A_{\text{yes}}, A_{\text{no}})$ is in P if there exists TM M and polynomial p, such that for any input $x \in \{0,1\}^n$, M halts in a most $O(p(n))$ steps and:

- (YES case) If $x \in A_{\text{yes}}$, M outputs 1.
- (NO case) If $x \in A_{\text{no}}$, M outputs 0.

Non-deterministic Polynomial-Time (NP)

Polynomial-Time (P)

Decision problem $A = (A_{\text{yes}}, A_{\text{no}})$ is in P if there exists TM M and polynomial p, such that for any input $x \in \{0,1\}^n$, M halts in a most $O(p(n))$ steps and:

- (YES case) If $x \in A_{\text{yes}}$, M outputs 1.
- (NO case) If $x \in A_{\text{no}}$, M outputs 0.

Non-deterministic Polynomial-Time (NP)

Decision problem $A = (A_{\text{yes}}, A_{\text{no}})$ is in NP if there exists TM M and polynomials p and q, such that for any input $x \in \{0,1\}^n$, M halts in a most $O(p(n))$ steps and:

- (YES case) If $x \in A_{\text{yes}}$, there exists proof $y \in \{0,1\}^{q(n)}$, such that $M(x,y) = 1$.
- (NO case) If $x \in A_{\text{no}}$, for all proofs $y \in \{0,1\}^{q(n)}$, $M(x,y) = 0$.

Observe: $\text{P} \subseteq \text{NP}$.

Non-deterministic Polynomial-Time (NP)

Polynomial-Time (P)

Decision problem $A = (A_{\text{yes}}, A_{\text{no}})$ is in P if there exists TM M and polynomial p, such that for any input $x \in \{0,1\}^n$, M halts in a most $O(p(n))$ steps and:

- (YES case) If $x \in A_{\text{yes}}$, M outputs 1.
- (NO case) If $x \in A_{\text{no}}$, M outputs 0.

Non-deterministic Polynomial-Time (NP)

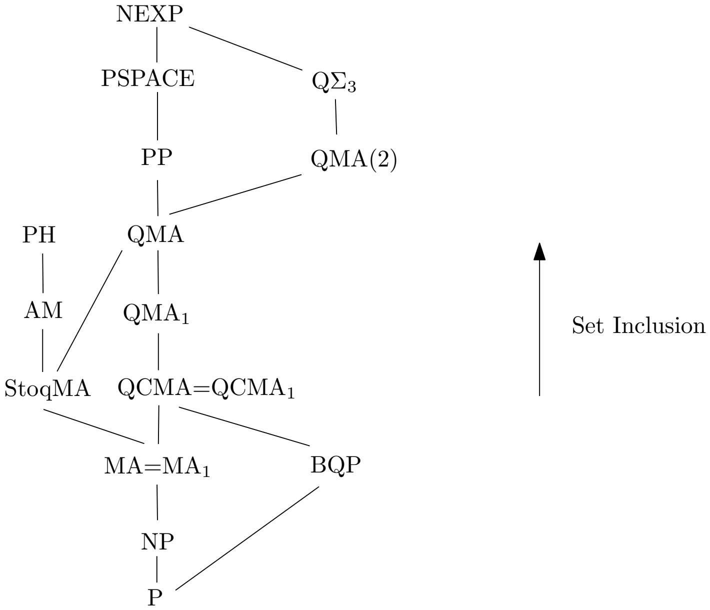
Decision problem $A = (A_{\text{yes}}, A_{\text{no}})$ is in NP if there exists TM M and polynomials p and q, such that for any input $x \in \{0,1\}^n$, M halts in a most $O(p(n))$ steps and:

- (YES case) If $x \in A_{\text{yes}}$, there exists proof $y \in \{0,1\}^{q(n)}$, such that $M(x,y) = 1$.
- (NO case) If $x \in A_{\text{no}}$, for all proofs $y \in \{0,1\}^{q(n)}$, $M(x,y) = 0$.

Observe: $\text{P} \subseteq \text{NP}$.

Note: FACTOR $\in$ NP (given "proof" $y \in \mathbb{Z}$, can efficiently check if $y \leq t$ and $x \mod y = 0$).

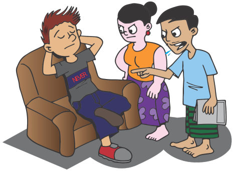
Preview


NEXP
PSPACE
QΣ3
PP
QMA(2)
PH
QMA
AM
QMA1
StoqMA
QCMA=QCMA1
MA=MA1
BQP
NP
P
Set Inclusion

Outline

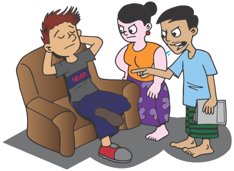
Reductions

Moral code of complexity theorists: Let someone else solve your problem



Reductions

Moral code of complexity theorists: Let someone else solve your problem



(Many-one) reduction

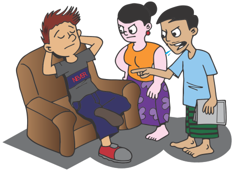
A reduction from $A = (A_{\text{yes}}, A_{\text{no}})$ to $B = (B_{\text{yes}}, B_{\text{no}})$, denoted $A \leq B$, is a TM M, s.t. for any input $x \in \{0,1\}^*$,

- if $x \in A_{\text{yes}}$, then $M(x) \in B_{\text{yes}}$.
- if $x \in A_{\text{no}}$, then $M(x) \in B_{\text{no}}$.

If M runs in poly-time, we say the reduction is poly-time, and write $A \leq_p B$.

Reductions

Moral code of complexity theorists: Let someone else solve your problem



(Many-one) reduction

A reduction from $A = (A_{\text{yes}}, A_{\text{no}})$ to $B = (B_{\text{yes}}, B_{\text{no}})$, denoted $A \leq B$, is a TM M, s.t. for any input $x \in \{0,1\}^*$,

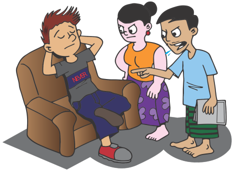
- if $x \in A_{\text{yes}}$, then $M(x) \in B_{\text{yes}}$.
- if $x \in A_{\text{no}}$, then $M(x) \in B_{\text{no}}$.

If M runs in poly-time, we say the reduction is poly-time, and write $A \leq_p B$.

- Implication: If $A \leq_p B$, then if $B \in \text{P} \Rightarrow A \in \text{P}$.

Reductions

Moral code of complexity theorists: Let someone else solve your problem



(Many-one) reduction

A reduction from $A = (A_{\text{yes}}, A_{\text{no}})$ to $B = (B_{\text{yes}}, B_{\text{no}})$, denoted $A \leq B$, is a TM M, s.t. for any input $x \in \{0,1\}^*$,

- if $x \in A_{\text{yes}}$, then $M(x) \in B_{\text{yes}}$.
- if $x \in A_{\text{no}}$, then $M(x) \in B_{\text{no}}$.

If M runs in poly-time, we say the reduction is poly-time, and write $A \leq_p B$.

- Implication: If $A \leq_p B$, then if $B \in \mathrm{P} \Rightarrow A \in \mathrm{P}$.
- Exercise 2: Show that MULTIPLY reduces to ADD $= \left\{(x_1, \ldots, x_k, t) \in \mathbb{Z}^{k+1} \mid k \geq 0 \text{ and } \sum_{i=1}^{k} x_k \leq t\right\}$.
  Is your reduction poly-time?

NP-complete problems

"Strongest/hardest" problems in NP

Formally:

- $B = (B_{\text{yes}}, B_{\text{no}})$ is NP-hard if for all $A = (A_{\text{yes}}, A_{\text{no}}) \in \text{NP}$, $A \leq_p B$.
  - Implication: $B \in \text{P} \Rightarrow \text{P} = \text{NP}$.

NP-complete problems

"Strongest/hardest" problems in NP

Formally:

- $B = (B_{\text{yes}}, B_{\text{no}})$ is NP-hard if for all $A = (A_{\text{yes}}, A_{\text{no}}) \in \text{NP}$, $A \leq_p B$.
  - Implication: $B \in \text{P} \Rightarrow \text{P} = \text{NP}$.
- B is NP-complete if B is NP-hard and $B \in \text{NP}$.
  - Implication: B "characterizes" the power of NP.

NP-complete problems

"Strongest/hardest" problems in NP

Formally:

- $B = (B_{\text{yes}}, B_{\text{no}})$ is NP-hard if for all $A = (A_{\text{yes}}, A_{\text{no}}) \in \text{NP}$, $A \leq_p B$.
  - Implication: $B \in \text{P} \Rightarrow \text{P} = \text{NP}$.
- B is NP-complete if B is NP-hard and $B \in \text{NP}$.
  - Implication: B "characterizes" the power of NP.
- Cook-Levin Theorem: 3-SAT is NP-complete [Cook 1971, Levin 1973]

3-SAT

Input: Boolean formula $\phi : \{0,1\}^n \rightarrow \{0,1\}$ in "3-Conjunctive Normal Form (3-CNF)", e.g.

$$\phi = (x_1 \vee x_2 \vee \overline{x_3}) \wedge (x_4 \vee \overline{x_1} \vee \overline{x_9}) \cdots (\overline{x_1} \vee x_5 \vee \overline{x_2})$$

Output: Is there a "satisfying assignment", i.e. $\exists x \in \{0,1\}^n$ such that $\phi(x) = 1$?

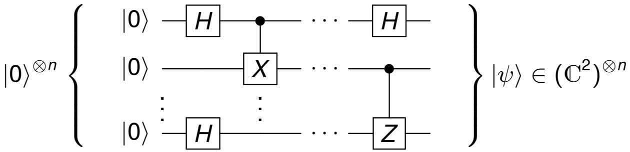
Exercise 3: Show that 3-SAT is NP-hard even if each variable x_i appears at most 3 times in ϕ.

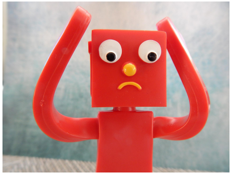
Exercise 4: What is the complexity of 3-SAT if each variable x_i appears exactly 3 times in ϕ? (Hint: Google "Hall's marriage theorem".)

Outline

Outline

Quantumly: Work with poly(n)-size quantum circuit implementing n-qubit unitaries U, e.g.





What happened to our beloved TMs?

The computational model

What computational model to use for quantum complexity theory?

- Idea 1: Use "quantum Turing machines"...

The computational model

What computational model to use for quantum complexity theory?

- Idea 1: Use "quantum Turing machines"...

- Idea 2: Use "poly-size" quantum circuits?

The computational model

What computational model to use for quantum complexity theory?

- Idea 1: Use "quantum Turing machines"...

- Idea 2: Use "poly-size" quantum circuits?
  - Exercise 5. If 3-SAT formula ϕ is satisfiable, $\exists$ poly-size circuit computing x with $\phi(x) = 1$.

The computational model

What computational model to use for quantum complexity theory?

- Idea 1: Use "quantum Turing machines"...

- Idea 2: Use "poly-size" quantum circuits?
  - Exercise 5. If 3-SAT formula ϕ is satisfiable, $\exists$ poly-size circuit computing x with $\phi(x) = 1$.
  - Problem: Even if poly-size circuit exists, can be hard to find it!
    If the poly-size circuit is hard to find, not very useful for solving problems ☹

The computational model

What computational model to use for quantum complexity theory?

- Idea 1: Use "quantum Turing machines"...

- Idea 2: Use "poly-size" quantum circuits?
  - Exercise 5. If 3-SAT formula ϕ is satisfiable, $\exists$ poly-size circuit computing x with $\phi(x) = 1$.
  - Problem: Even if poly-size circuit exists, can be hard to find it!
    If the poly-size circuit is hard to find, not very useful for solving problems ☺
  - Solution: Use "poly-time uniformly generated" circuits.

Remark: All quantum circuits in this lecture are sequences of 1- and 2-qubit gates.

P-uniform quantum circuit family

A family of quantum circuits $\{Q_n\}$ is P-uniform if there exists a poly-time TM M, which given as input 1^n, outputs a classical description of Q_n via a sequence of 1- and 2-qubit gates.

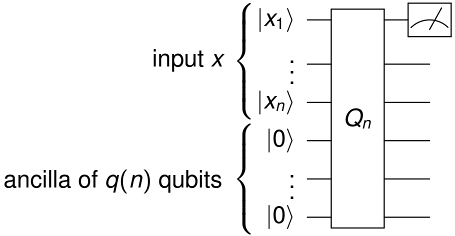
Exercise 6: Why does M get 1^n as input, instead of n written in binary?

Henceforth: Use P-uniform quantum circuit families, not TMs.

Bounded-error quantum polynomial-time (BQP)

Promise problem $\mathbb{A} = (A_{\mathrm{yes}}, A_{\mathrm{no}}, A_{\mathrm{inv}}) \in \mathrm{BQP}$ if $\exists$ P-uniform quantum circuit family $\{Q_n\}$ and polynomial q as below. The first output qubit of Q_n is measured in the standard basis and returned. For any input $x \in \{0,1\}^*$:

- (YES case) If $x \in A_{\mathrm{yes}}$, then Q_n outputs 1 with probability at least $2/3$.
- (NO case) If $x \in A_{\mathrm{no}}$, then Q_n outputs 1 with probability at most $1/3$.
- (Invalid case) If $x \in A_{\mathrm{inv}}$, Q_n outputs 0 or 1 arbitrarily.



Caution

Quantum complexity classes typically promise classes

[2] Best strategy: Run circuit in parallel, take majority vote of output answers, apply Chernoff bound.

Caution

Quantum complexity classes typically promise classes

- Decision problem: All inputs are "valid", i.e. A_{yes} and A_{no} partition $\{0, 1\}^*$.

[2] Best strategy: Run circuit in parallel, take majority vote of output answers, apply Chernoff bound.

Caution

Quantum complexity classes typically promise classes

- Decision problem: All inputs are "valid", i.e. A_{yes} and A_{no} partition $\{0,1\}^*$.
- Promise problem: Not all inputs are "valid", i.e. $A_{\text{inv}} = \{0,1\}^* \setminus (A_{\text{yes}} \cup A_{\text{no}})$.
  - Intuition: $\text{poly}(n)$ runs of quantum circuit cannot distinguish[2] YES vs NO thresholds like

$$\frac{1}{2}+\frac{1}{2^n} \quad \text{versus} \quad \frac{1}{2}-\frac{1}{2^n}.$$

[2] Best strategy: Run circuit in parallel, take majority vote of output answers, apply Chernoff bound.

Caution

Quantum complexity classes typically promise classes

- Decision problem: All inputs are "valid", i.e. A_{yes} and A_{no} partition $\{0,1\}^*$.
- Promise problem: Not all inputs are "valid", i.e. $A_{\text{inv}} = \{0,1\}^* \setminus (A_{\text{yes}} \cup A_{\text{no}})$.
  - Intuition: $\text{poly}(n)$ runs of quantum circuit cannot distinguish[2] YES vs NO thresholds like

$$\frac{1}{2}+\frac{1}{2^n} \quad \text{versus} \quad \frac{1}{2}-\frac{1}{2^n}.$$

  - Exercise 7: Chernoff bound has "exponential scaling". Why does it not suffice above?

[2]Best strategy: Run circuit in parallel, take majority vote of output answers, apply Chernoff bound.

Outline

The Pikachu of BQP

Linear system solving:

- Input: Invertible $A \in \mathbb{C}^{N\times N}$ and target vector $\mathbf{b} \in \mathbb{C}^N$
- Output: $\mathbf{x} \in \mathbb{C}^N$ such that $A\mathbf{x} = \mathbf{b}$.

What is the complexity of linear system solving?

The Pikachu of BQP

Linear system solving:

- Input: Invertible $A \in \mathbb{C}^{N \times N}$ and target vector $\mathbf{b} \in \mathbb{C}^N$
- Output: $\mathbf{x} \in \mathbb{C}^N$ such that $A\mathbf{x} = \mathbf{b}$.

What is the complexity of linear system solving?

- A and $\mathbf{x}$ given explicitly in matrix form $\Rightarrow \mathbf{x} = A^{-1}\mathbf{b}$ classically in time $\text{poly}(N)$
  - Exercise 8: Is linear system solving thus in P?

The Pikachu of BQP

Linear system solving:

- Input: Invertible $A \in \mathbb{C}^{N\times N}$ and target vector $\mathbf{b} \in \mathbb{C}^N$
- Output: $\mathbf{x} \in \mathbb{C}^N$ such that $A\mathbf{x} = \mathbf{b}$.

What is the complexity of linear system solving?

- A and $\mathbf{x}$ given explicitly in matrix form $\Rightarrow$ $\mathbf{x} = A^{-1}\mathbf{b}$ classically in time $\mathrm{poly}(N)$
  - Exercise 8: Is linear system solving thus in P?
- A represented "succinctly" via "query-access" and $\mathbf{b}$ given via quantum circuit?

Matrix inversion problem (MI)

Input:

- $O(1)$-sparse row-computable invertible Hermitian matrix[3] $A \in \mathbb{C}^{N\times N}$
- A specified via polylog$N$-time TM M which, given row index $r \in [N]$ of A, outputs entries of row r of A

[3] Technically, need condition number $\kappa(A)$ to satisfy $\kappa^{-1}(A) \preceq A \preceq I$ with $\kappa(A) \in \text{polylog}(N)$.

Matrix inversion problem (MI)

Input:

- $O(1)$-sparse row-computable invertible Hermitian matrix[3] $A \in \mathbb{C}^{N\times N}$
- A specified via polylog$N$-time TM M which, given row index $r \in [N]$ of A, outputs entries of row r of A

Output: Let $|x\rangle \propto A^{-1}|0^N\rangle$ be a unit vector, and $\Pi = |1\rangle\langle 1|$ a projector onto the first qubit of $|x\rangle$. Then:

- (YES case) If $\langle x|\Pi|x\rangle \geq 2/3$, output YES.
- (NO case) If $\langle x|\Pi|x\rangle \leq 1/3$, output NO.
- (Invalid case) Else, output YES or NO arbitrarily.

[3] Technically, need condition number $\kappa(A)$ to satisfy $\kappa^{-1}(A) \preceq A \preceq I$ with $\kappa(A) \in \text{polylog}(N)$.

Matrix inversion problem (MI)

Input:

- $O(1)$-sparse row-computable invertible Hermitian matrix[3] $A \in \mathbb{C}^{N\times N}$
- A specified via polylog$N$-time TM M which, given row index $r \in [N]$ of A, outputs entries of row r of A

Output: Let $|x\rangle \propto A^{-1}|0^N\rangle$ be a unit vector, and $\Pi = |1\rangle\langle 1|$ a projector onto the first qubit of $|x\rangle$. Then:

- (YES case) If $\langle x|\Pi|x\rangle \geq 2/3$, output YES.
- (NO case) If $\langle x|\Pi|x\rangle \leq 1/3$, output NO.
- (Invalid case) Else, output YES or NO arbitrarily.

Theorem [Harrow, Hassidim, Lloyd, 2008]

MI is BQP-complete under poly-time many-one reductions.

Proof steps:

1. MI $\in \mathrm{BQP}$.
2. MI is BQP-hard.

[3] Technically, need condition number $\kappa(A)$ to satisfy $\kappa^{-1}(A) \preceq A \preceq I$ with $\kappa(A) \in \mathrm{polylog}(N)$.

Matrix inversion problem (MI)

Input:

- $O(1)$-sparse row-computable invertible Hermitian matrix[3] $A \in \mathbb{C}^{N \times N}$
- A specified via polylog$N$-time TM M which, given row index $r \in [N]$ of A, outputs entries of row r of A

Output: Let $|x\rangle \propto A^{-1}|0^N\rangle$ be a unit vector, and $\Pi = |1\rangle\langle 1|$ a projector onto the first qubit of $|x\rangle$. Then:

- (YES case) If $\langle x|\Pi|x\rangle \geq 2/3$, output YES.
- (NO case) If $\langle x|\Pi|x\rangle \leq 1/3$, output NO.
- (Invalid case) Else, output YES or NO arbitrarily.

Theorem [Harrow, Hassidim, Lloyd, 2008]

MI is BQP-complete under poly-time many-one reductions.

Proof steps:

1. MI $\in$ BQP.
2. MI is BQP-hard.

[3] Technically, need condition number $\kappa(A)$ to satisfy $\kappa^{-1}(A)I \preceq A \preceq I$ with $\kappa(A) \in$ polylog(N).

Outline

MI $\in$ BQP

Goal: Given sparse Hermitian A and poly-size circuit for $|b\rangle$, want to compute unit vector $|x\rangle \propto A^{-1}|b\rangle$.

Idea: To compute A^{-1}, *coherently invert* each eigenvalue of A via Quantum Phase Estimation (QPE).

MI $\in$ BQP

Goal: Given sparse Hermitian A and poly-size circuit for $|b\rangle$, want to compute unit vector $|x\rangle \propto A^{-1}|b\rangle$.

Idea: To compute A^{-1}, *coherently invert* each eigenvalue of A via Quantum Phase Estimation (QPE).

Notation: Spectral decomposition $A = \sum_i \lambda_i |\psi_i\rangle\langle\psi_i|$.

Framework: Eigenvalue surgery

1. Eigenvalue extraction (via Hamiltonian simulation and Quantum Phase Estimation (QPE))
2. Eigenvalue processing (done classically, coherently)
3. Eigenvalue reinsertion (via postselection)

MI $\in$ BQP

Goal: Given sparse Hermitian A and poly-size circuit for $|b\rangle$, want to compute unit vector $|x\rangle \propto A^{-1}|b\rangle$.

Idea: To compute A^{-1}, *coherently invert* each eigenvalue of A via Quantum Phase Estimation (QPE).

Notation: Spectral decomposition $A = \sum_i \lambda_i |\psi_i\rangle\langle\psi_i|$.

Framework: Eigenvalue surgery

1. Eigenvalue extraction (via Hamiltonian simulation and Quantum Phase Estimation (QPE))
2. Eigenvalue processing (done classically, coherently)
3. Eigenvalue reinsertion (via postselection)

Hamiltonian simulation

Question: Why is quantum dynamics *unitary*?

Hamiltonian simulation

Question: Why is quantum dynamics *unitary*?

(Time-independent) Schrödinger equation

Time evolution of any n-qubit system governed by Hermitian matrix $H \in \mathcal{L}(\mathbb{C}^2)^{\otimes n}$, called a Hamiltonian:

$$i\frac{d|\psi\rangle}{dt} = H|\psi\rangle$$

Hamiltonian simulation

Question: Why is quantum dynamics *unitary*?

(Time-independent) Schrödinger equation

Time evolution of any n-qubit system governed by Hermitian matrix $H \in \mathcal{L}(\mathbb{C}^2)^{\otimes n}$, called a Hamiltonian:

$$i\frac{d|\psi\rangle}{dt} = H|\psi\rangle \quad \xrightarrow{solve} \quad |\psi_t\rangle = e^{-iHt}|\psi_0\rangle \quad (\leftarrow \text{ unitary!})$$

Hamiltonian simulation

Question: Why is quantum dynamics *unitary*?

(Time-independent) Schrödinger equation

Time evolution of any n-qubit system governed by Hermitian matrix $H \in \mathcal{L}(\mathbb{C}^2)^{\otimes n}$, called a Hamiltonian:

$$i\frac{d|\psi\rangle}{dt} = H|\psi\rangle \quad \xrightarrow{solve} \quad |\psi_t\rangle = e^{-iHt}|\psi_0\rangle \quad (\leftarrow \text{ unitary!})$$

Hamiltonian simulation [Low, Chuang 2017]

Given d-sparse H, simulation time $t \geq 0$, and $\epsilon > 0$, can simulate e^{iHt} up to error ϵ and success probability at least $1 - 2\epsilon$ in time[a]

$$O\left(td\,\|H\|_{\max} + \frac{\log(1/\epsilon)}{\log\log(1/\epsilon)}\right).$$

[a] Query complexity. Gate complexity has $O(n)$ overhead.

MI $\in$ BQP

Goal: Given sparse Hermitian A and poly-size circuit for $|b\rangle$, want to compute unit vector $|x\rangle \propto A^{-1}|b\rangle$.

Idea: To compute A^{-1}, *coherently invert* each eigenvalue of A via Quantum Phase Estimation (QPE).

Notation: Spectral decomposition $A = \sum_i \lambda_i |\psi_i\rangle\langle\psi_i|$.

Framework: Eigenvalue surgery

1. Eigenvalue extraction (via Hamiltonian simulation and Quantum Phase Estimation (QPE))
2. Eigenvalue processing (done classically, coherently)
3. Eigenvalue reinsertion (via postselection)

Quantum Phase Estimation (QPE)

- Consider Hermitian H with spectral decomposition $H = \sum_j \lambda_j |\psi_j\rangle\langle\psi_j|$.
- Consider spectral decomposition of unitary:

$$U = e^{iH} =$$

Quantum Phase Estimation (QPE)

- Consider Hermitian H with spectral decomposition $H = \sum_j \lambda_j |\psi_j\rangle\langle\psi_j|$.
- Consider spectral decomposition of unitary:

$$U = e^{iH} = \sum_j e^{i\lambda_j} |\psi_j\rangle\langle\psi_j|.$$

Quantum Phase Estimation (QPE)

- Consider Hermitian H with spectral decomposition $H = \sum_j \lambda_j |\psi_j\rangle\langle\psi_j|$.
- Consider spectral decomposition of unitary:

$$U = e^{iH} = \sum_j e^{i\lambda_j} |\psi_j\rangle\langle\psi_j|.$$

- Goal: Given eigenvector $|\psi_j\rangle$, precision parameter k, want to compute λ_j to k bits of precision.

Quantum Phase Estimation algorithm (QPE)

Given precision k, and ability to efficiently compute controlled-U^{2^K} for $1 \leq K \leq k$, can map

$$|0^k\rangle|\psi_j\rangle \mapsto |\widetilde{\lambda_j}\rangle|\psi_j\rangle$$

Quantum Phase Estimation (QPE)

- Consider Hermitian H with spectral decomposition $H = \sum_j \lambda_j |\psi_j\rangle\langle\psi_j|$.
- Consider spectral decomposition of unitary:

$$U = e^{iH} = \sum_j e^{i\lambda_j} |\psi_j\rangle\langle\psi_j|.$$

- Goal: Given eigenvector $|\psi_j\rangle$, precision parameter k, want to compute λ_j to k bits of precision.

Quantum Phase Estimation algorithm (QPE)

Given precision k, and ability to efficiently compute controlled-U^{2^K} for $1 \leq K \leq k$, can map

$$|0^k\rangle|\psi_j\rangle \mapsto |\widetilde{\lambda}_j\rangle|\psi_j\rangle \quad \Rightarrow \quad |0^k\rangle \sum_j \alpha_j |\psi_j\rangle \mapsto \sum_j \alpha_j |\widetilde{\lambda}_j\rangle|\psi_j\rangle,$$

where $\widetilde{\lambda}_j$ is λ_j up to k bits.

Exercise 9a: Given n-qubit unitary U, can we efficiently compute U^{2^n} in general?

MI $\in$ BQP

Goal: Given sparse Hermitian A and poly-size circuit for $|b\rangle$, want to compute unit vector $|x\rangle \propto A^{-1}|b\rangle$.

Idea: To compute A^{-1}, *coherently invert* each eigenvalue of A via Quantum Phase Estimation (QPE).

Notation: Spectral decomposition $A = \sum_i \lambda_i |\psi_i\rangle\langle\psi_i|$.

Framework: Eigenvalue surgery

1. Eigenvalue extraction (via Hamiltonian simulation and Quantum Phase Estimation (QPE))
2. Eigenvalue processing (done classically, coherently)
3. Eigenvalue reinsertion (via postselection)

Step 1: Eigenvalue extraction (recall $A = \sum_i \lambda_i |\psi_i\rangle\langle\psi_i|$)

- Prepare target state

$$|b\rangle = \sum_{j=1}^{N} \alpha_j |\psi_j\rangle \in \mathbb{C}^N,$$

where $|\psi_j\rangle$ are the eigenvectors of A with eigenvalues λ_j.

Step 1: Eigenvalue extraction (recall $A = \sum_i \lambda_i |\psi_i\rangle\langle\psi_i|$)

- Prepare target state

$$|b\rangle = \sum_{j=1}^{N} \alpha_j |\psi_j\rangle \in \mathbb{C}^N,$$

where $|\psi_j\rangle$ are the eigenvectors of A with eigenvalues λ_j.

- Apply QPE (for unitary e^{iA}) with an n-qubit ancilla to our state $|b\rangle$ to obtain

$$\sum_{j=1}^{N} \alpha_j |\lambda_j\rangle |\psi_j\rangle \in (\mathbb{C}^2)^{\otimes n} \otimes \mathbb{C}^N.$$

Step 1: Eigenvalue extraction (recall $A = \sum_i \lambda_i |\psi_i\rangle\langle\psi_i|$)

- Prepare target state

$$|b\rangle = \sum_{j=1}^{N} \alpha_j |\psi_j\rangle \in \mathbb{C}^N,$$

where $|\psi_j\rangle$ are the eigenvectors of A with eigenvalues λ_j.

- Apply QPE (for unitary e^{iA}) with an n-qubit ancilla to our state $|b\rangle$ to obtain

$$\sum_{j=1}^{N} \alpha_j |\lambda_j\rangle |\psi_j\rangle \in (\mathbb{C}^2)^{\otimes n} \otimes \mathbb{C}^N.$$

Step 2: Eigenvalue processing

- Conditioned on the first register, rotate a new single-qubit ancilla as follows:

$$\sum_{j=1}^{N} \alpha_j |\lambda_j\rangle |\psi_j\rangle \left(\sqrt{1 - \frac{1}{\lambda_j^2 \kappa^2(A)}} |0\rangle + \left(\frac{1}{\lambda_j \kappa(A)} \right) |1\rangle \right) \in (\mathbb{C}^2)^{\otimes n} \otimes \mathbb{C}^N \otimes \mathbb{C}^2.$$

Exercise 9b: Google “condition number”, learn about it.

Exercise 10: Assume $\|A\|_\infty = 1$. Show $1/\kappa(A) \leq 1/(\lambda_j \kappa(A)) \leq 1$. Thus, amplitudes above well-defined.

Step 2: Eigenvalue processing

- Conditioned on the first register, rotate a new single-qubit ancilla as follows:

$$\sum_{j=1}^{N} \alpha_j |\lambda_j\rangle|\psi_j\rangle \left(\sqrt{1 - \frac{1}{\lambda_j^2 \kappa^2(A)}} |0\rangle + \left(\frac{1}{\lambda_j \kappa(A)} \right) |1\rangle \right) \in (\mathbb{C}^2)^{\otimes n} \otimes \mathbb{C}^N \otimes \mathbb{C}^2.$$

Step 2: Eigenvalue processing

- Conditioned on the first register, rotate a new single-qubit ancilla as follows:

$$\sum_{j=1}^{N} \alpha_j |\lambda_j\rangle|\psi_j\rangle \left(\sqrt{1 - \frac{1}{\lambda_j^2 \kappa^2(A)}} |0\rangle + \left(\frac{1}{\lambda_j \kappa(A)} \right) |1\rangle \right) \in (\mathbb{C}^2)^{\otimes n} \otimes \mathbb{C}^N \otimes \mathbb{C}^2.$$

Step 3: Eigenvalue reinsertion

- Measure third register in standard basis, postselect on outcome 1, discard third register:

$$\sum_{j=1}^{N} \alpha_j \left(\frac{1}{\lambda_j} \right) |\psi_j\rangle \propto A^{-1}|b\rangle \in \mathbb{C}^N.$$

Exercise 11. Prove that probability of obtaining outcome 1 is at least $1/\kappa^2(A)$.

Exercise 12. What is the expected number of repetitions for postselection to succeed? Can we improve this with amplitude amplification?

Runtime

If we run QPE to get additive inverse poly error for phases, runtime is

$$\widetilde{O}(\kappa(A)(T_b + s^2 \log^2(N)))$$

for T_b the number of gates to prepare $|b\rangle$, N the dimension of A, and $\log N$ the number of qubits.

Implication:

- When $\kappa(A), T_b, s \in \text{polylog}(N)$, exponentially faster than classically solving the entire $N \times N$ system.
- For definition of MI, suffices to obtain MI $\in \mathrm{BQP}$.

Exercise 13**. Although the quantum algorithm can give exponential speedups, why is it incorrect to directly compare it to classical linear system solvers?

Matrix inversion problem (MI)

Input:

- $O(1)$-sparse row-computable invertible Hermitian matrix[4] $A \in \mathbb{C}^{N\times N}$
- A specified via poly-time TM M which, given row index $r \in [N]$ of A, outputs entries of row r of A

Output: Let $|x\rangle \propto A^{-1}|0^N\rangle$ be a unit vector, and $\Pi = |1\rangle\langle 1|$ a projector onto the first qubit of $|x\rangle$. Then:

- (YES case) If $\langle x|\Pi|x\rangle \geq 2/3$, output YES.
- (NO case) If $\langle x|\Pi|x\rangle \leq 1/3$, output NO.
- (Invalid case) Else, output YES or NO arbitrarily.

Theorem [Harrow, Hassidim, Lloyd, 2008]

MI is BQP-complete under poly-time many-one reductions.

Proof steps:

1. MI $\in$ BQP.
2. MI is BQP-hard.

[4] Technically, need condition number $\kappa(A)$ to satisfy $\kappa^{-1}(A)I \preceq A \preceq I$ with $\kappa(A) \in \text{polylog}(N)$.

Outline

MI is BQP-hard

Goal: Show that any BQP computation V poly-time reducible to an instance A of MI.

Starting point: Let $V = V_m \cdots V_1$ be a BQP circuit on n qubits, $N = 2^n$. Assume WLOG m is power of 2.

Problem: Need to tie matrix inverse with action of V.

MI is BQP-hard

Goal: Show that any BQP computation V poly-time reducible to an instance A of MI.

Starting point: Let $V = V_m \cdots V_1$ be a BQP circuit on n qubits, $N = 2^n$. Assume WLOG m is power of 2.

Problem: Need to tie matrix inverse with action of V.

Idea:

- Recall Maclaurin series $\frac{1}{1-x} = \sum_{l=0}^{\infty} x^l$ for $|x| < 1$.
- We could apply this to any normal matrix U with $\|U\|_\infty < 1$ to get

$$(I - U)^{-1} = \sum_{l=0}^{\infty} U^l.$$

MI is BQP-hard

Goal: Show that any BQP computation V poly-time reducible to an instance A of MI.

Starting point: Let $V = V_m \cdots V_1$ be a BQP circuit on n qubits, $N = 2^n$. Assume WLOG m is power of 2.

Problem: Need to tie matrix inverse with action of V.

Idea:

- Recall Maclaurin series $\frac{1}{1-x} = \sum_{l=0}^{\infty} x^l$ for $|x| < 1$.
- We could apply this to any normal matrix U with $\|U\|_\infty < 1$ to get

$$(I - U)^{-1} = \sum_{l=0}^{\infty} U^l.$$

- What would be great: Normal matrix U acting something like

$$U^k|0^n\rangle \approx V_k \cdots V_1|0^n\rangle.$$

- What would be great: Normal matrix U acting something like

$$U^{k}|0^n\rangle \approx V_k \cdots V_1|0^n\rangle.$$

- Define:

$$U = \sum_{t=0}^{m-1} |t+1\rangle\langle t| \otimes V_{t+1} + \sum_{t=m}^{2m-1} |t+1 \bmod 2m\rangle\langle t| \otimes V^{\dagger}_{2m-t} \in \mathcal{U}((\mathbb{C}^2)^{\otimes \log m} \otimes (\mathbb{C}^2)^{\otimes n}),$$

Exercise 12: Check that U is unitary.

Exercise 13: Check that $U^m|0^{\log m}\rangle|0^n\rangle = |m\rangle V|0^n\rangle$.

Implication: Measuring first qubit of second register of $U^m|0^{\log m}\rangle|0^n\rangle$ simulates measuring output qubit of V!

- We could apply this to any normal matrix U with $\|U\|_\infty < 1$ to get $(I-U)^{-1} = \sum_{l=0} U^l$.
- Define $U = \sum_{t=0}^{m-1} |t+1\rangle\langle t| \otimes V_{t+1} + \sum_{t=m}^{2m-1} |t+1 \bmod 2m\rangle\langle t| \otimes V_{2m-t}^\dagger \in \mathcal{U}((\mathbb{C}^2)^{\otimes \log m} \otimes (\mathbb{C}^2)^{\otimes n})$,

- We could apply this to any normal matrix U with $\|U\|_\infty < 1$ to get $(I-U)^{-1} = \sum_{l=0} U^l$.
- Define $U = \sum_{t=0}^{m-1} |t+1\rangle\langle t| \otimes V_{t+1} + \sum_{t=m}^{2m-1} |t+1 \bmod 2m\rangle\langle t| \otimes V_{2m-t}^\dagger \in \mathcal{U}((\mathbb{C}^2)^{\otimes \log m} \otimes (\mathbb{C}^2)^{\otimes n})$,
- Define $A = I - U$. Then,

$$|x\rangle \quad \propto \quad A^{-1}|0^{\log m+n}\rangle$$

- We could apply this to any normal matrix U with $\|U\|_\infty < 1$ to get $(I-U)^{-1} = \sum_{l=0} U^l$.
- Define $U = \sum_{t=0}^{m-1} |t+1\rangle\langle t| \otimes V_{t+1} + \sum_{t=m}^{2m-1} |t+1 \bmod 2m\rangle\langle t| \otimes V_{2m-t}^\dagger \in \mathcal{U}((\mathbb{C}^2)^{\otimes \log m} \otimes (\mathbb{C}^2)^{\otimes n})$,
- Define $A = I - U$. Then,

$$
\begin{aligned}
|x\rangle &\propto A^{-1}|0^{\log m+n}\rangle \\
&= (I-U)^{-1}|0^{\log m+n}\rangle
\end{aligned}
$$

- We could apply this to any normal matrix U with $\|U\|_\infty < 1$ to get $(I-U)^{-1} = \sum_{l=0} U^l$.
- Define $U = \sum_{t=0}^{m-1} |t+1\rangle\langle t| \otimes V_{t+1} + \sum_{t=m}^{2m-1} |t+1 \bmod 2m\rangle\langle t| \otimes V_{2m-t}^\dagger \in \mathcal{U}((\mathbb{C}^2)^{\otimes \log m} \otimes (\mathbb{C}^2)^{\otimes n})$,
- Define $A = I - U$. Then,

$$\begin{aligned} |x\rangle &\propto A^{-1}|0^{\log m+n}\rangle \\ &= (I-U)^{-1}|0^{\log m+n}\rangle \\ &\propto \sum_{l=0}^{\infty} U^l|0\rangle^{\log m}|0^n\rangle \end{aligned}$$

- We could apply this to any normal matrix U with $\|U\|_\infty < 1$ to get $(I-U)^{-1} = \sum_{l=0} U^l$.
- Define $U = \sum_{t=0}^{m-1} |t+1\rangle\langle t| \otimes V_{t+1} + \sum_{t=m}^{2m-1} |t+1 \bmod 2m\rangle\langle t| \otimes V_{2m-t}^\dagger \in \mathcal{U}((\mathbb{C}^2)^{\otimes \log m} \otimes (\mathbb{C}^2)^{\otimes n})$,
- Define $A = I - U$. Then,

$$
\begin{aligned}
|x\rangle &\propto A^{-1}|0^{\log m+n}\rangle \\
&= (I-U)^{-1}|0^{\log m+n}\rangle \\
&\propto \sum_{l=0}^{\infty} U^l |0\rangle^{\log m}|0^n\rangle \\
&\propto |0\rangle|0^n\rangle + |1\rangle V_1|0^n\rangle + \cdots + |m\rangle V_m \cdots V_1|0^n\rangle.
\end{aligned}
$$

- We could apply this to any normal matrix U with $\|U\|_\infty < 1$ to get $(I-U)^{-1} = \sum_{l=0} U^l$.
- Define $U = \sum_{t=0}^{m-1} |t+1\rangle\langle t| \otimes V_{t+1} + \sum_{t=m}^{2m-1} |t+1 \bmod 2m\rangle\langle t| \otimes V_{2m-t}^\dagger \in \mathcal{U}((\mathbb{C}^2)^{\otimes \log m} \otimes (\mathbb{C}^2)^{\otimes n})$,
- Define $A = I - U$. Then,

$$
\begin{aligned}
|x\rangle &\propto A^{-1}|0^{\log m+n}\rangle \\
&= (I-U)^{-1}|0^{\log m+n}\rangle \\
&\propto \sum_{l=0}^{\infty} U^l |0\rangle^{\log m}|0^n\rangle \\
&\propto |0\rangle|0^n\rangle + |1\rangle V_1|0^n\rangle + \cdots + |m\rangle V_m \cdots V_1|0^n\rangle.
\end{aligned}
$$

- Implication:
  - Measuring first register gives $|m\rangle$ with probability $\approx 1/(m+1)$.

- We could apply this to any normal matrix U with $\|U\|_\infty < 1$ to get $(I-U)^{-1} = \sum_{l=0} U^l$.
- Define $U = \sum_{t=0}^{m-1} |t+1\rangle\langle t| \otimes V_{t+1} + \sum_{t=m}^{2m-1} |t+1 \bmod 2m\rangle\langle t| \otimes V_{2m-t}^\dagger \in \mathcal{U}((\mathbb{C}^2)^{\otimes \log m} \otimes (\mathbb{C}^2)^{\otimes n})$,
- Define $A = I - U$. Then,

$$\begin{aligned} |x\rangle &\propto A^{-1}|0^{\log m+n}\rangle \\ &= (I-U)^{-1}|0^{\log m+n}\rangle \\ &\propto \sum_{l=0}^{\infty} U^l |0\rangle^{\log m}|0^n\rangle \\ &\propto |0\rangle|0^n\rangle + |1\rangle V_1|0^n\rangle + \cdots + |m\rangle V_m \cdots V_1|0^n\rangle. \end{aligned}$$

- Implication:
  - Measuring first register gives $|m\rangle$ with probability $\approx 1/(m+1)$.
  - Postselecting on $|m\rangle$, measuring second register reveals BQP circuit V's output.

- We could apply this to any normal matrix U with $\|U\|_\infty < 1$ to get $(I-U)^{-1} = \sum_{l=0} U^l$.
- Define $U = \sum_{t=0}^{m-1} |t+1\rangle\langle t| \otimes V_{t+1} + \sum_{t=m}^{2m-1} |t+1 \bmod 2m\rangle\langle t| \otimes V_{2m-t}^\dagger \in \mathcal{U}((\mathbb{C}^2)^{\otimes \log m} \otimes (\mathbb{C}^2)^{\otimes n})$,
- Define $A = I - U$. Then,

$$
\begin{aligned}
|x\rangle &\propto A^{-1}|0^{\log m+n}\rangle \\
&= (I-U)^{-1}|0^{\log m+n}\rangle \\
&\propto \sum_{l=0}^{\infty} U^l |0\rangle^{\log m}|0^n\rangle \\
&\propto |0\rangle|0^n\rangle + |1\rangle V_1|0^n\rangle + \cdots + |m\rangle V_m \cdots V_1|0^n\rangle.
\end{aligned}
$$

- Implication:
  - Measuring first register gives $|m\rangle$ with probability $\approx 1/(m+1)$.
  - Postselecting on $|m\rangle$, measuring second register reveals BQP circuit V's output.

Exercise 14: I cheated slightly on one of the lines above (regarding $|x\rangle$) — where did I cheat?

- We could apply this to any normal matrix U with $\|U\|_\infty < 1$ to get $(I-U)^{-1} = \sum_{l=0} U^l$.
- Define $U = \sum_{t=0}^{m-1} |t+1\rangle\langle t| \otimes V_{t+1} + \sum_{t=m}^{2m-1} |t+1 \bmod 2m\rangle\langle t| \otimes V_{2m-t}^\dagger \in \mathcal{U}((\mathbb{C}^2)^{\otimes \log m} \otimes (\mathbb{C}^2)^{\otimes n})$,
- Define $A = I - U$. Then,

$$\begin{aligned} |x\rangle &\propto A^{-1}|0^{\log m+n}\rangle \\ &= (I-U)^{-1}|0^{\log m+n}\rangle \\ &\propto \sum_{l=0}^{\infty} U^l |0\rangle^{\log m}|0^n\rangle \\ &\propto |0\rangle|0^n\rangle + |1\rangle V_1|0^n\rangle + \cdots + |m\rangle V_m \cdots V_1|0^n\rangle. \end{aligned}$$

- Implication:
  - Measuring first register gives $|m\rangle$ with probability $\approx 1/(m+1)$.
  - Postselecting on $|m\rangle$, measuring second register reveals BQP circuit V's output.

Exercise 14: I cheated slightly on one of the lines above (regarding $|x\rangle$) — where did I cheat?

Exercise 15: I cheated less slightly somewhere else on this slide. Where did I make a bigger boo boo?

Final exercises for MI

Construction *almost* works, but for 3 issues to check:

1. A must be $O(1)$-sparse (by def of MI).

   Exercise 16: Check that U, and thus A, are $O(1)$-sparse.

Final exercises for MI

Construction *almost* works, but for 3 issues to check:

1. A must be $O(1)$-sparse (by def of MI).

   Exercise 16: Check that U, and thus A, are $O(1)$-sparse.

2. MI needs YES case and NO case thresholds of $2/3$ vs $1/3$ for BQP. The current construction will give $2/(3(m+1))$ vs $1/(3(m+1))$.

   Exercise 17: Modify the construction to boost the YES/NO thresholds to $2/3$ and $1/3$, respectively.

Final exercises for MI

Construction *almost* works, but for 3 issues to check:

1. A must be $O(1)$-sparse (by def of MI).

   Exercise 16: Check that U, and thus A, are $O(1)$-sparse.

2. MI needs YES case and NO case thresholds of $2/3$ vs $1/3$ for BQP. The current construction will give $2/(3(m+1))$ vs $1/(3(m+1))$.

   Exercise 17: Modify the construction to boost the YES/NO thresholds to $2/3$ and $1/3$, respectively.

3. Our current choice of A is not necessarily invertible, since $\|U\|_\infty = 1$. (Maclaurin series does not apply.)

   Exercise 18: Consider first $A = I - \frac{1}{2}U$. Show that A is invertible and has $\kappa(A) \in O(1)$. Where will this construction nevertheless fail in the analysis?

Final exercises for MI

Construction *almost* works, but for 3 issues to check:

1. A must be $O(1)$-sparse (by def of MI).

   Exercise 16: Check that U, and thus A, are $O(1)$-sparse.

2. MI needs YES case and NO case thresholds of $2/3$ vs $1/3$ for BQP. The current construction will give $2/(3(m+1))$ vs $1/(3(m+1))$.

   Exercise 17: Modify the construction to boost the YES/NO thresholds to $2/3$ and $1/3$, respectively.

3. Our current choice of A is not necessarily invertible, since $\|U\|_\infty = 1$. (Maclaurin series does not apply.)

   Exercise 18: Consider first $A = I - \frac{1}{2}U$. Show that A is invertible and has $\kappa(A) \in O(1)$. Where will this construction nevertheless fail in the analysis?

   Exercise 19: Consider finally $A = I - e^{-1/m}U$. Show that A is invertible, has $\kappa(A) \in O(m) \in \text{polylog}(N)$. Show that this choice avoids the problem from Exercise 18.

Final exercises for MI

Construction *almost* works, but for 3 issues to check:

1. A must be $O(1)$-sparse (by def of MI).

   Exercise 16: Check that U, and thus A, are $O(1)$-sparse.

2. MI needs YES case and NO case thresholds of $2/3$ vs $1/3$ for BQP. The current construction will give $2/(3(m+1))$ vs $1/(3(m+1))$.

   Exercise 17: Modify the construction to boost the YES/NO thresholds to $2/3$ and $1/3$, respectively.

3. Our current choice of A is not necessarily invertible, since $\|U\|_\infty = 1$. (Maclaurin series does not apply.)

   Exercise 18: Consider first $A = I - \frac{1}{2}U$. Show that A is invertible and has $\kappa(A) \in O(1)$. Where will this construction nevertheless fail in the analysis?

   Exercise 19: Consider finally $A = I - e^{-1/m}U$. Show that A is invertible, has $\kappa(A) \in O(m) \in \text{polylog}(N)$. Show that this choice avoids the problem from Exercise 18.

4. I cheated again. There is a 4th issue — A must be Hermitian. But I will spare you these details.

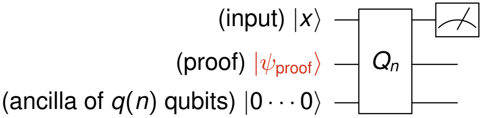
Outline

QMA

Quantum Merlin-Arthur (QMA)

Promise problem $\mathbb{A} = (A_{\text{yes}}, A_{\text{no}}, A_{\text{inv}}) \in$ QMA if $\exists$ P-uniform quantum circuit family $\{Q_n\}$ and polynomials p,q:

- (YES case) If $x \in A_{\text{yes}}$, $\exists$ proof $|\psi_{\text{proof}}\rangle \in (\mathbb{C}^2)^{\otimes p(n)}$, such that Q_n accepts with probability at least $2/3$.
- (NO case) If $x \in A_{\text{no}}$, then $\forall$ proofs $|\psi_{\text{proof}}\rangle \in (\mathbb{C}^2)^{\otimes p(n)}$, Q_n accepts with probability at most $1/3$.
- (Invalid case) If $x \in A_{\text{inv}}$, Q_n may accept or reject arbitrarily.



Error reduction for QMA

Weak error reduction (the "obvious" type)

- Idea: Given $\text{poly}(n)$ copies of $|\psi_{\text{proof}}\rangle$, repeat verification $\text{poly}(n)$ times and take majority vote

Error reduction for QMA

Weak error reduction (the "obvious" type)

- Idea: Given $\text{poly}(n)$ copies of $|\psi_{\text{proof}}\rangle$, repeat verification $\text{poly}(n)$ times and take majority vote
- Caution: In NO case, no guarantee proof is of form $|\psi_{\text{proof}}\rangle \otimes |\psi_{\text{proof}}\rangle \otimes \cdots \otimes |\psi_{\text{proof}}\rangle$!

Error reduction for QMA

Weak error reduction (the "obvious" type)

- Idea: Given $\text{poly}(n)$ copies of $|\psi_{\text{proof}}\rangle$, repeat verification $\text{poly}(n)$ times and take majority vote
- Caution: In NO case, no guarantee proof is of form $|\psi_{\text{proof}}\rangle \otimes |\psi_{\text{proof}}\rangle \otimes \cdots \otimes |\psi_{\text{proof}}\rangle$!
- Achieves completeness $1 - 2^{-\text{poly}(n)}$ versus soundness $2^{-\text{poly}(n)}$.

Error reduction for QMA

Weak error reduction (the "obvious" type)

- Idea: Given $\text{poly}(n)$ copies of $|\psi_{\text{proof}}\rangle$, repeat verification $\text{poly}(n)$ times and take majority vote
- Caution: In NO case, no guarantee proof is of form $|\psi_{\text{proof}}\rangle \otimes |\psi_{\text{proof}}\rangle \otimes \cdots \otimes |\psi_{\text{proof}}\rangle$!
- Achieves completeness $1 - 2^{-\text{poly}(n)}$ versus soundness $2^{-\text{poly}(n)}$.

Problem: Blows up proof size (by a polynomial)

Error reduction for QMA

Weak error reduction (the "obvious" type)

- Idea: Given $\text{poly}(n)$ copies of $|\psi_{\text{proof}}\rangle$, repeat verification $\text{poly}(n)$ times and take majority vote
- Caution: In NO case, no guarantee proof is of form $|\psi_{\text{proof}}\rangle \otimes |\psi_{\text{proof}}\rangle \otimes \cdots \otimes |\psi_{\text{proof}}\rangle$!
- Achieves completeness $1 - 2^{-\text{poly}(n)}$ versus soundness $2^{-\text{poly}(n)}$.

Problem: Blows up proof size (by a polynomial)

Question: Can we do it with just *one* copy of $|\psi_{\text{proof}}\rangle$?

Error reduction for QMA

Weak error reduction (the "obvious" type)

- Idea: Given $\text{poly}(n)$ copies of $|\psi_{\text{proof}}\rangle$, repeat verification $\text{poly}(n)$ times and take majority vote
- Caution: In NO case, no guarantee proof is of form $|\psi_{\text{proof}}\rangle \otimes |\psi_{\text{proof}}\rangle \otimes \cdots \otimes |\psi_{\text{proof}}\rangle$!
- Achieves completeness $1 - 2^{-\text{poly}(n)}$ versus soundness $2^{-\text{poly}(n)}$.

Problem: Blows up proof size (by a polynomial)

Question: Can we do it with just *one* copy of $|\psi_{\text{proof}}\rangle$?

Obstacle: No-cloning theorem says we cannot *copy* $|\psi_{\text{proof}}\rangle$...

Marriot-Watrous strong error reduction

- Set $i = 0$.
- Do while $i \leq N$:
  - (Run verification Q_n) Run Q_n and measure output qubit to obtain bit y_i. Set $i = i + 1$.
  - (Run Q_n in reverse) Run $Q_n^\dagger$ and measure whether input "resets" to x and ancillae to $|0 \cdots 0\rangle$. If yes, set $y_i = 1$, else set $y_i = 0$. Set $i = i + 1$.
- (Postprocessing) If the number of indices $i \in \{0, \ldots, N-1\}$ such that $y_i = y_{i+1}$ is at least $N/2$, accept. Otherwise, reject.

Outline

Remember this?

(Time-independent) Schrödinger equation

Time evolution of any n-qubit system governed by Hermitian matrix $H \in \mathcal{L}(\mathbb{C}^2)^{\otimes n}$, called a Hamiltonian:

$$i\frac{d|\psi\rangle}{dt} = H|\psi\rangle \quad \xrightarrow{\text{solve}} \quad |\psi_t\rangle = e^{-iHt}|\psi_0\rangle \quad (\leftarrow \text{ unitary!})$$

Remember this?

(Time-independent) Schrödinger equation

Time evolution of any n-qubit system governed by Hermitian matrix $H \in \mathcal{L}(\mathbb{C}^2)^{\otimes n}$, called a Hamiltonian:

$$i\frac{d|\psi\rangle}{dt} = H|\psi\rangle \quad \xrightarrow{\text{solve}} \quad |\psi_t\rangle = e^{-iHt}|\psi_0\rangle \quad (\leftarrow \text{ unitary!})$$

Question: What kind of Hamiltonians H appear in nature?

k-local Hamiltonian

k-local Hamiltonian

An n-qubit Hermitian operator $H = \sum_i H_i \in \mathcal{L}((\mathbb{C}^2)^{\otimes n})$, where

- each H_i is a $2^k \times 2^k$ matrix for $k \in O(1)$, i.e. a quantum constraint,

k-local Hamiltonian

k-local Hamiltonian

An n-qubit Hermitian operator $H = \sum_i H_i \in \mathcal{L}((\mathbb{C}^2)^{\otimes n})$, where

- each H_i is a $2^k \times 2^k$ matrix for $k \in O(1)$, i.e. a quantum constraint,
- smallest eigenvalue $\lambda_{\min}(H)$ is ground state energy,

k-local Hamiltonian

k-local Hamiltonian

An n-qubit Hermitian operator $H = \sum_i H_i \in \mathcal{L}((\mathbb{C}^2)^{\otimes n})$, where

- each H_i is a $2^k \times 2^k$ matrix for $k \in O(1)$, i.e. a quantum constraint,
- smallest eigenvalue $\lambda_{\min}(H)$ is ground state energy,
- the eigenvector $|\psi\rangle \in (\mathbb{C}^2)^{\otimes n}$ corresponding to $\lambda_{\min}(H)$ is ground state.

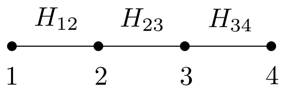
k-local Hamiltonian

k-local Hamiltonian

An n-qubit Hermitian operator $H = \sum_i H_i \in \mathcal{L}((\mathbb{C}^2)^{\otimes n})$, where

- each H_i is a $2^k \times 2^k$ matrix for $k \in O(1)$, i.e. a quantum constraint,
- smallest eigenvalue $\lambda_{\min}(H)$ is ground state energy,
- the eigenvector $|\psi\rangle \in (\mathbb{C}^2)^{\otimes n}$ corresponding to $\lambda_{\min}(H)$ is ground state.

Example. Let $H_{ij} = X_i \otimes X_j + Y_i \otimes Y_j + Z_i \otimes Z_j \in \mathcal{L}(\mathbb{C}^4)$.



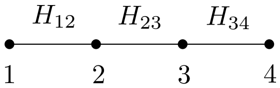
k-local Hamiltonian

k-local Hamiltonian

An n-qubit Hermitian operator $H = \sum_i H_i \in \mathcal{L}((\mathbb{C}^2)^{\otimes n})$, where

- each H_i is a $2^k \times 2^k$ matrix for $k \in O(1)$, i.e. a quantum constraint,
- smallest eigenvalue $\lambda_{\min}(H)$ is ground state energy,
- the eigenvector $|\psi\rangle \in (\mathbb{C}^2)^{\otimes n}$ corresponding to $\lambda_{\min}(H)$ is ground state.

Example. Let $H_{ij} = X_i \otimes X_j + Y_i \otimes Y_j + Z_i \otimes Z_j \in \mathcal{L}(\mathbb{C}^4)$.



Then, $H = H_{12} \otimes I_{34} + I_1 \otimes H_{23} \otimes I_4 + I_{12} \otimes H_{34} \in \mathcal{L}(\mathbb{C}^{16})$.

Quantum constraint satisfaction

k-local Hamiltonian problem (k-LH)

- Input: k-local Hamiltonian H on n qubits, thresholds $0 \leq \alpha \leq \beta$ s.t. $|\alpha - \beta| \geq 1/\operatorname{poly}(n)$
- Promise: $\lambda_{\min}(H) \leq \alpha$ or $\lambda_{\min}(H) \geq \beta$
- Output: Decide whether $\lambda_{\min}(H) \leq \alpha$ or $\lambda_{\min}(H) \geq \beta$

- Canonical QMA-complete problem!
- Motivation: Show superfluid helium video

https://www.youtube.com/watch?v=2Z6UJbwxBZI

Selected history

- "Quantum Cook-Levin Theorem": 5-LH is QMA-complete [Kitaev, 1999]

Selected history

- "Quantum Cook-Levin Theorem": 5-LH is QMA-complete [Kitaev, 1999]
- 2-LH is QMA-complete (via perturbation theory gadgets) [Kempe, Kitaev, Regev, 2004]

Selected history

- "Quantum Cook-Levin Theorem": 5-LH is QMA-complete [Kitaev, 1999]
- 2-LH is QMA-complete (via perturbation theory gadgets) [Kempe, Kitaev, Regev, 2004]
- QMA-complete on 2D square lattice of qubits [Oliveira, Terhal 2005]

Selected history

- “Quantum Cook-Levin Theorem”: 5-LH is QMA-complete [Kitaev, 1999]
- 2-LH is QMA-complete (via perturbation theory gadgets) [Kempe, Kitaev, Regev, 2004]
- QMA-complete on 2D square lattice of qubits [Oliveira, Terhal 2005]
- QMA-complete in 1D (!) for local dimension 12 [Aharonov, Gottesman, Irani, Kempe 2009]

Selected history

- "Quantum Cook-Levin Theorem": 5-LH is QMA-complete [Kitaev, 1999]
- 2-LH is QMA-complete (via perturbation theory gadgets) [Kempe, Kitaev, Regev, 2004]
- QMA-complete on 2D square lattice of qubits [Oliveira, Terhal 2005]
- QMA-complete in 1D (!) for local dimension 12 [Aharonov, Gottesman, Irani, Kempe 2009]
- "Quantum NEXP"-complete for 1D, translation-invariant systems [Gottesman, Irani, 2010]

Selected history

- “Quantum Cook-Levin Theorem”: 5-LH is QMA-complete [Kitaev, 1999]
- 2-LH is QMA-complete (via perturbation theory gadgets) [Kempe, Kitaev, Regev, 2004]
- QMA-complete on 2D square lattice of qubits [Oliveira, Terhal 2005]
- QMA-complete in 1D (!) for local dimension 12 [Aharonov, Gottesman, Irani, Kempe 2009]
- “Quantum NEXP”-complete for 1D, translation-invariant systems [Gottesman, Irani, 2010]
- 4-chotomy theorem: either P, NP-complete, StoqMA-complete, QMA-complete [Cubitt, Montanaro 2013], [Bravyi, Hastings 2014]

Selected history

- "Quantum Cook-Levin Theorem": 5-LH is QMA-complete [Kitaev, 1999]
- 2-LH is QMA-complete (via perturbation theory gadgets) [Kempe, Kitaev, Regev, 2004]
- QMA-complete on 2D square lattice of qubits [Oliveira, Terhal 2005]
- QMA-complete in 1D (!) for local dimension 12 [Aharonov, Gottesman, Irani, Kempe 2009]
- "Quantum NEXP"-complete for 1D, translation-invariant systems [Gottesman, Irani, 2010]
- 4-chotomy theorem: either P, NP-complete, StoqMA-complete, QMA-complete [Cubitt, Montanaro 2013], [Bravyi, Hastings 2014]

Variants:

- PSPACE-complete for $|\alpha - \beta| \geq 1/\exp(n)$ [Fefferman, Lin, 2016]

Selected history

- "Quantum Cook-Levin Theorem": 5-LH is QMA-complete [Kitaev, 1999]
- 2-LH is QMA-complete (via perturbation theory gadgets) [Kempe, Kitaev, Regev, 2004]
- QMA-complete on 2D square lattice of qubits [Oliveira, Terhal 2005]
- QMA-complete in 1D (!) for local dimension 12 [Aharonov, Gottesman, Irani, Kempe 2009]
- "Quantum NEXP"-complete for 1D, translation-invariant systems [Gottesman, Irani, 2010]
- 4-chotomy theorem: either P, NP-complete, StoqMA-complete, QMA-complete [Cubitt, Montanaro 2013], [Bravyi, Hastings 2014]

Variants:

- PSPACE-complete for $|\alpha - \beta| \geq 1/\exp(n)$ [Fefferman, Lin, 2016]
- QMA-hard for $|\alpha - \beta| \in \Omega(1)$?

Quantum PCP conjecture! (see [Aharonov, Arad, Vidick, 2013] for survey)

Kitaev's quantum Cook-Levin theorem

Goal: Map U to instance $(H, \alpha, \beta, |\psi\rangle)$ of LH such that $\beta - \alpha \geq 1/\operatorname{poly}(n)$ and

$$\begin{aligned} \text{if } U \text{ accepts } x \quad &\Longrightarrow \quad \lambda_{\min}(H) \leq \alpha \\ \text{if } U \text{ rejects } x \quad &\Longrightarrow \quad \lambda_{\min}(H) \geq \beta \end{aligned}$$

Kitaev's quantum Cook-Levin theorem

Goal: Map U to instance $(H, \alpha, \beta, |\psi\rangle)$ of LH such that $\beta - \alpha \geq 1/\operatorname{poly}(n)$ and

$$\begin{aligned} \text{if } U \text{ accepts } x &\implies \lambda_{\min}(H) \leq \alpha \\ \text{if } U \text{ rejects } x &\implies \lambda_{\min}(H) \geq \beta \end{aligned}$$

- Let $U = U_m \cdots U_1$ be a QMA circuit verifying proof $|\psi_{\text{proof}}\rangle$.
- Design local terms H_i to force ground state to be history state:

$$|\psi_{\text{hist}}\rangle = \frac{1}{\sqrt{m+1}} \sum_{t=0}^{m} U_t \cdots U_1 |\psi_{\text{proof}}\rangle_A |0 \cdots 0\rangle_B |t\rangle_C$$

A: proof register B: ancilla register C: clock register

Feynman-Kitaev circuit-to-Hamiltonian construction

$$|\psi_{\text{hist}}\rangle = \frac{1}{\sqrt{m+1}} \sum_{t=0}^{m} U_t \cdots U_1 |\psi_{\text{proof}}\rangle_A |0\cdots 0\rangle_B |t\rangle_C$$

Feynman-Kitaev circuit-to-Hamiltonian construction

$$|\psi_{\text{hist}}\rangle = \frac{1}{\sqrt{m+1}}\sum_{t=0}^{m} U_t \cdots U_1 |\psi_{\text{proof}}\rangle_A |0\cdots 0\rangle_B |t\rangle_C$$

Define $H = H_{\text{in}} + H_{\text{out}} + H_{\text{prop}} + H_{\text{stab}}$ such that

H_{in}: Correct ancilla initialization at time $t = 0$ $\Rightarrow$ $\langle\psi_{\text{hist}}|H_{\text{in}}|\psi_{\text{hist}}\rangle = 0$

Feynman-Kitaev circuit-to-Hamiltonian construction

$$|\psi_{\text{hist}}\rangle = \frac{1}{\sqrt{m+1}} \sum_{t=0}^{m} U_t \cdots U_1 |\psi_{\text{proof}}\rangle_A |0\cdots 0\rangle_B |t\rangle_C$$

Define $H = H_{\text{in}} + H_{\text{out}} + H_{\text{prop}} + H_{\text{stab}}$ such that

H_{in}:	Correct ancilla initialization at time $t=0$	$\Rightarrow$	$\langle\psi_{\text{hist}}\|H_{\text{in}}\|\psi_{\text{hist}}\rangle = 0$
H_{prop}:	Gate U_t applied at time t	$\Rightarrow$	$\langle\psi_{\text{hist}}\|H_{\text{prop}}\|\psi_{\text{hist}}\rangle = 0$

Feynman-Kitaev circuit-to-Hamiltonian construction

$$|\psi_{\text{hist}}\rangle = \frac{1}{\sqrt{m+1}}\sum_{t=0}^{m} U_t \cdots U_1 |\psi_{\text{proof}}\rangle_A |0\cdots 0\rangle_B |t\rangle_C$$

Define $H = H_{\text{in}} + H_{\text{out}} + H_{\text{prop}} + H_{\text{stab}}$ such that

H_{in}:	Correct ancilla initialization at time $t = 0$	$\Rightarrow$	$\langle\psi_{\text{hist}}\|H_{\text{in}}\|\psi_{\text{hist}}\rangle = 0$
H_{prop}:	Gate U_t applied at time t	$\Rightarrow$	$\langle\psi_{\text{hist}}\|H_{\text{prop}}\|\psi_{\text{hist}}\rangle = 0$
H_{stab}:	Clock register C encoded correctly in unary	$\Rightarrow$	$\langle\psi_{\text{hist}}\|H_{\text{out}}\|\psi_{\text{hist}}\rangle = 0$

Feynman-Kitaev circuit-to-Hamiltonian construction

$$|\psi_{\text{hist}}\rangle = \frac{1}{\sqrt{m+1}} \sum_{t=0}^{m} U_t \cdots U_1 |\psi_{\text{proof}}\rangle_A |0\cdots 0\rangle_B |t\rangle_C$$

Define $H = H_{\text{in}} + H_{\text{out}} + H_{\text{prop}} + H_{\text{stab}}$ such that

H_{in}:	Correct ancilla initialization at time $t = 0$	$\Rightarrow$	$\langle\psi_{\text{hist}}\|H_{\text{in}}\|\psi_{\text{hist}}\rangle = 0$
H_{prop}:	Gate U_t applied at time t	$\Rightarrow$	$\langle\psi_{\text{hist}}\|H_{\text{prop}}\|\psi_{\text{hist}}\rangle = 0$
H_{stab}:	Clock register C encoded correctly in unary	$\Rightarrow$	$\langle\psi_{\text{hist}}\|H_{\text{out}}\|\psi_{\text{hist}}\rangle = 0$
H_{out}:	Penalize rejecting computation U at time $t = m$	$\Rightarrow$	$\langle\psi_{\text{hist}}\|H_{\text{out}}\|\psi_{\text{hist}}\rangle \sim \frac{1-\Pr(U \text{ accepts } x)}{\text{poly}(m)}$

Feynman-Kitaev circuit-to-Hamiltonian construction

$$|\psi_{\text{hist}}\rangle = \frac{1}{\sqrt{m+1}}\sum_{t=0}^{m} U_t\cdots U_1|\psi_{\text{proof}}\rangle_A|0\cdots0\rangle_B|t\rangle_C$$

Define $H = H_{\text{in}} + H_{\text{out}} + H_{\text{prop}} + H_{\text{stab}}$ such that

H_{in}:	Correct ancilla initialization at time $t=0$	$\Rightarrow$	$\langle\psi_{\text{hist}}\|H_{\text{in}}\|\psi_{\text{hist}}\rangle = 0$
H_{prop}:	Gate U_t applied at time t	$\Rightarrow$	$\langle\psi_{\text{hist}}\|H_{\text{prop}}\|\psi_{\text{hist}}\rangle = 0$
H_{stab}:	Clock register C encoded correctly in unary	$\Rightarrow$	$\langle\psi_{\text{hist}}\|H_{\text{out}}\|\psi_{\text{hist}}\rangle = 0$
H_{out}:	Penalize rejecting computation U at time $t=m$	$\Rightarrow$	$\langle\psi_{\text{hist}}\|H_{\text{out}}\|\psi_{\text{hist}}\rangle \sim \frac{1-\Pr(U \text{ accepts } x)}{\text{poly}(m)}$

$$H_{\text{in}} = I_A \otimes (I - |0\cdots0\rangle\langle0\cdots0|)_B \otimes |0\rangle\langle0|_C$$

Feynman-Kitaev circuit-to-Hamiltonian construction

$$|\psi_{\text{hist}}\rangle = \frac{1}{\sqrt{m+1}}\sum_{t=0}^{m} U_t \cdots U_1 |\psi_{\text{proof}}\rangle_A |0\cdots 0\rangle_B |t\rangle_C$$

Define $H = H_{\text{in}} + H_{\text{out}} + H_{\text{prop}} + H_{\text{stab}}$ such that

H_{in}:	Correct ancilla initialization at time $t = 0$	$\Rightarrow$	$\langle\psi_{\text{hist}}\|H_{\text{in}}\|\psi_{\text{hist}}\rangle = 0$
H_{prop}:	Gate U_t applied at time t	$\Rightarrow$	$\langle\psi_{\text{hist}}\|H_{\text{prop}}\|\psi_{\text{hist}}\rangle = 0$
H_{stab}:	Clock register C encoded correctly in unary	$\Rightarrow$	$\langle\psi_{\text{hist}}\|H_{\text{out}}\|\psi_{\text{hist}}\rangle = 0$
H_{out}:	Penalize rejecting computation U at time $t = m$	$\Rightarrow$	$\langle\psi_{\text{hist}}\|H_{\text{out}}\|\psi_{\text{hist}}\rangle \sim \frac{1-\Pr(U \text{ accepts } x)}{\text{poly}(m)}$

$$\begin{aligned} H_{\text{in}} &= I_A \otimes (I - |0\cdots 0\rangle\langle 0\cdots 0|)_B \otimes |0\rangle\langle 0|_C \\ H_{\text{out}} &= I_A \otimes |0\rangle\langle 0|_{B_1} \otimes |m\rangle\langle m|_C. \end{aligned}$$

Feynman-Kitaev circuit-to-Hamiltonian construction

$$|\psi_{\text{hist}}\rangle = \frac{1}{\sqrt{m+1}} \sum_{t=0}^{m} U_t \cdots U_1 |\psi_{\text{proof}}\rangle_A |0\cdots 0\rangle_B |t\rangle_C$$

Define $H = H_{\text{in}} + H_{\text{out}} + H_{\text{prop}} + H_{\text{stab}}$ such that

H_{in}:	Correct ancilla initialization at time $t = 0$	$\Rightarrow$	$\langle\psi_{\text{hist}}\|H_{\text{in}}\|\psi_{\text{hist}}\rangle = 0$
H_{prop}:	Gate U_t applied at time t	$\Rightarrow$	$\langle\psi_{\text{hist}}\|H_{\text{prop}}\|\psi_{\text{hist}}\rangle = 0$
H_{stab}:	Clock register C encoded correctly in unary	$\Rightarrow$	$\langle\psi_{\text{hist}}\|H_{\text{out}}\|\psi_{\text{hist}}\rangle = 0$
H_{out}:	Penalize rejecting computation U at time $t = m$	$\Rightarrow$	$\langle\psi_{\text{hist}}\|H_{\text{out}}\|\psi_{\text{hist}}\rangle \sim \frac{1-\Pr(U \text{ accepts } x)}{\text{poly}(m)}$

$$\begin{aligned} H_{\text{in}} &= I_A \otimes (I - |0\cdots 0\rangle\langle 0\cdots 0|)_B \otimes |0\rangle\langle 0|_C \\ H_{\text{out}} &= I_A \otimes |0\rangle\langle 0|_{B_1} \otimes |m\rangle\langle m|_C. \end{aligned}$$

Question: How to check time propagation, i.e. U_t applied at time t?

The propagation term

$$|\psi_{\text{hist}}\rangle = \frac{1}{\sqrt{m+1}} \sum_{t=0}^{m} U_t \cdots U_1 |\psi_{\text{proof}}\rangle_A |0\cdots 0\rangle_B |t\rangle_C$$

Goal: Define H_{prop} so that if U_t applied at time $t \Rightarrow \langle\psi_{\text{hist}}|H_{\text{prop}}|\psi_{\text{hist}}\rangle = 0$.

The propagation term

$$|\psi_{\text{hist}}\rangle = \frac{1}{\sqrt{m+1}}\sum_{t=0}^{m} U_t\cdots U_1|\psi_{\text{proof}}\rangle_A|0\cdots 0\rangle_B|t\rangle_C$$

Goal: Define H_{prop} so that if U_t applied at time $t \Rightarrow \langle\psi_{\text{hist}}|H_{\text{prop}}|\psi_{\text{hist}}\rangle = 0$.

Define for each $t \in \{0,\ldots,m-1\}$:

$$H_{\text{prop},t} = -U_t \otimes |t\rangle\langle t-1|_C - U_t^\dagger \otimes |t-1\rangle\langle t|_C + I \otimes |t-1\rangle\langle t-1|_C + I \otimes |t\rangle\langle t|_C,$$

Why does this work?

The propagation term

$$|\psi_{\text{hist}}\rangle = \frac{1}{\sqrt{m+1}}\sum_{t=0}^{m} U_t\cdots U_1|\psi_{\text{proof}}\rangle_A|0\cdots 0\rangle_B|t\rangle_C$$

Goal: Define H_{prop} so that if U_t applied at time $t \Rightarrow \langle\psi_{\text{hist}}|H_{\text{prop}}|\psi_{\text{hist}}\rangle = 0$.

Define for each $t \in \{0,\ldots,m-1\}$:

$$H_{\text{prop},t} = -U_t\otimes|t\rangle\langle t-1|_C - U_t^\dagger\otimes|t-1\rangle\langle t|_C + I\otimes|t-1\rangle\langle t-1|_C + I\otimes|t\rangle\langle t|_C,$$

Why does this work?

$$\sum_{t=0}^{m} H_{\text{prop},t}$$

The propagation term

$$|\psi_{\text{hist}}\rangle = \frac{1}{\sqrt{m+1}}\sum_{t=0}^{m} U_t\cdots U_1|\psi_{\text{proof}}\rangle_A|0\cdots 0\rangle_B|t\rangle_C$$

Goal: Define H_{prop} so that if U_t applied at time $t \Rightarrow \langle\psi_{\text{hist}}|H_{\text{prop}}|\psi_{\text{hist}}\rangle = 0$.

Define for each $t \in \{0,\ldots,m-1\}$:

$$H_{\text{prop},t} = -U_t \otimes |t\rangle\langle t-1|_C - U_t^\dagger \otimes |t-1\rangle\langle t|_C + I \otimes |t-1\rangle\langle t-1|_C + I \otimes |t\rangle\langle t|_C,$$

Why does this work?

$$\sum_{t=0}^{m} H_{\text{prop},t} \overset{\text{change of basis}}{\mapsto} I_{AB} \otimes \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & \cdots \\ -1 & 2 & -1 & 0 & 0 & \cdots \\ 0 & -1 & 2 & -1 & 0 & \cdots \\ 0 & 0 & -1 & 2 & -1 & \cdots \\ 0 & 0 & 0 & -1 & \ddots & \ddots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \ddots \end{pmatrix}_C$$

The propagation term

$$|\psi_{\text{hist}}\rangle = \frac{1}{\sqrt{m+1}} \sum_{t=0}^{m} U_t \cdots U_1 |\psi_{\text{proof}}\rangle_A |0\cdots 0\rangle_B |t\rangle_C$$

Goal: Define H_{prop} so that if U_t applied at time $t \Rightarrow \langle \psi_{\text{hist}} | H_{\text{prop}} | \psi_{\text{hist}} \rangle = 0$.

Define for each $t \in \{0, \ldots, m-1\}$:

$$H_{\text{prop},t} = -U_t \otimes |t\rangle\langle t-1|_C - U_t^\dagger \otimes |t-1\rangle\langle t|_C + I \otimes |t-1\rangle\langle t-1|_C + I \otimes |t\rangle\langle t|_C,$$

Why does this work?

$$\sum_{t=0}^{m} H_{\text{prop},t} \overset{\text{change of basis}}{\mapsto} I_{AB} \otimes \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & \cdots \\ -1 & 2 & -1 & 0 & 0 & \cdots \\ 0 & -1 & 2 & -1 & 0 & \cdots \\ 0 & 0 & -1 & 2 & -1 & \cdots \\ 0 & 0 & 0 & -1 & \ddots & \ddots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \ddots \end{pmatrix}_C \Rightarrow \text{Unique null state (w.r.t. } C\text{): } I_{AB} \otimes \sum_{t=0}^{m} |t\rangle_C$$

Correctness

Completeness: By design,

$$\langle\psi_{\text{hist}}|H_{\text{in}} + H_{\text{prop}} + H_{\text{out}} + H_{\text{stab}}|\psi_{\text{hist}}\rangle \sim 0 + 0 + 0 + \frac{1 - \Pr(U \text{ accepts } x)}{\text{poly}(m)} \sim \text{“small”}.$$

Correctness

Completeness: By design,

$$\langle\psi_{\text{hist}}|H_{\text{in}} + H_{\text{prop}} + H_{\text{out}} + H_{\text{stab}}|\psi_{\text{hist}}\rangle \sim 0 + 0 + 0 + \frac{1 - \Pr(U \text{ accepts } x)}{\text{poly}(m)} \sim \text{“small”}.$$

Soundness:

- Goal: Show $\lambda_{\min}(H_{\text{in}} + H_{\text{prop}} + H_{\text{out}} + H_{\text{stab}}) \geq$ “large”.
- Problem: $H_{\text{in}} + H_{\text{out}}$ and H_{prop} do not commute (i.e. cannot add $\lambda_{\min}(H_{\text{in}} + H_{\text{out}})$ and $\lambda_{\min}(H_{\text{prop}})$)!

Geometric Lemma

Let $A_1, A_2 \succeq 0$, and let v lower bound the minimum non-zero eigenvalues of both A_1 and A_2. Then,

$$\lambda_{\min}(A_1 + A_2) \geq 2v \sin^2 \frac{\angle(\mathrm{Null}(A_1), \mathrm{Null}(A_2))}{2},$$

where the angle between spaces $\mathcal{X}$ and $\mathcal{Y}$ is defined as

$$\angle(\mathcal{X}, \mathcal{Y}) := \arccos\left(\max_{\substack{|x\rangle \in \mathcal{X}, |y\rangle \in \mathcal{Y} \\ \| |x\rangle \|_2 = \| |y\rangle \|_2 = 1}} |\langle x|y\rangle| \right).$$

Recall:

- $\mathrm{Null}(H_{\mathrm{in}} + H_{\mathrm{out}})$ - correct initialization and correct input
- $\mathrm{Null}(H_{\mathrm{prop}})$ - correct time propagation

Outline

Wait... there's more than one definition "quantum NP"?

Wait... there's more than one definition "quantum NP"?

Wait... there's more than one definition "quantum NP"?

Here we go (named after Snow White's dwarves):

- (Doc) QMA

Wait... there's more than one definition "quantum NP"?

Here we go (named after Snow White's dwarves):

- (Doc) QMA
- (Bashful) QMA_1: QMA with perfect completeness

Wait... there's more than one definition "quantum NP"?

Here we go (named after Snow White's dwarves):

- (Doc) QMA
- (Bashful) QMA_1: QMA with perfect completeness
- (Happy) QCMA: QMA with classical proof

Wait... there's more than one definition "quantum NP"?

Here we go (named after Snow White's dwarves):

- (Doc) QMA
- (Bashful) QMA_1: QMA with perfect completeness
- (Happy) QCMA: QMA with classical proof
- (Grumpy) $\mathrm{QMA}(2)$: QMA with "unentangled" proof of form $|\psi_1\rangle \otimes |\psi_2\rangle$

Wait... there's more than one definition "quantum NP"?

Here we go (named after Snow White's dwarves):

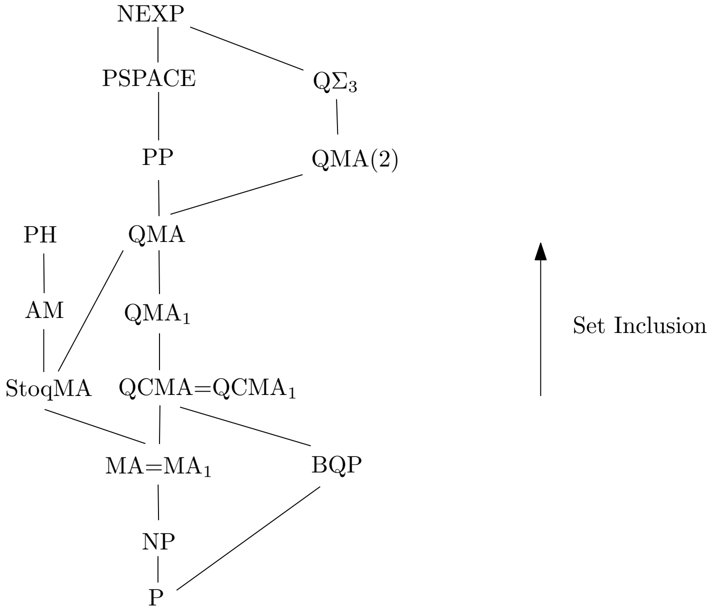
- (Doc) QMA
- (Bashful) QMA_1: QMA with perfect completeness
- (Happy) QCMA: QMA with classical proof
- (Grumpy) $\mathrm{QMA}(2)$: QMA with "unentangled" proof of form $|\psi_1\rangle \otimes |\psi_2\rangle$
- (Sneezy) NQP: Quantum TM accepts $x \in A_{\mathrm{yes}}$ in poly-time with probability > 0.
  (Equals coC$_=$P [Fenner, Green, Homer, Pruim, 1998].)

Wait... there's more than one definition "quantum NP"?

Here we go (named after Snow White's dwarves):

- (Doc) QMA
- (Bashful) QMA_1: QMA with perfect completeness
- (Happy) QCMA: QMA with classical proof
- (Grumpy) $\mathrm{QMA}(2)$: QMA with "unentangled" proof of form $|\psi_1\rangle \otimes |\psi_2\rangle$
- (Sneezy) NQP: Quantum TM accepts $x \in A_{\text{yes}}$ in poly-time with probability > 0.
  (Equals $\mathrm{coC_{=}P}$ [Fenner, Green, Homer, Pruim, 1998].)
- (Dopey) StoqMA: QMA with $\{|0\rangle, |+\rangle\}$ ancillae, classical gates, measurement in X basis

Relationships



Grumpy

QMA(2)

What does an "unentangled" proof $|\psi_1\rangle \otimes |\psi_2\rangle$ buy us?

QMA(2)

Promise problem $\mathbb{A} = (A_{\text{yes}}, A_{\text{no}}, A_{\text{inv}}) \in \text{QMA}(2)$ if there exists P-uniform quantum circuit family $\{Q_n\}$ s.t.:

- (YES) If $x \in A_{\text{yes}}$, $\exists$ proof $|\psi_1\rangle \otimes |\psi_2\rangle \in (\mathbb{C}^2)^{\otimes \text{poly}(n)} \otimes (\mathbb{C}^2)^{\otimes \text{poly}(n)}$, s.t. Q_n accepts w.p. $\geq 2/3$.
- (NO) If $x \in A_{\text{no}}$, then $\forall$ proofs $|\psi_1\rangle \otimes |\psi_2\rangle \in (\mathbb{C}^2)^{\otimes \text{poly}(n)} \otimes (\mathbb{C}^2)^{\otimes \text{poly}(n)}$, Q_n accepts w.p. $\leq 1/3$.
- (Invalid case) If $x \in A_{\text{inv}}$, Q_n may accept or reject arbitrarily.

Defined as QMA(k) for k parties by [Kobayashi, Matsumoto, Yamakami 2003]

Sad state of affairs: QMA $\subseteq \text{QMA}(2) \subseteq \text{Q}\Sigma_3 \subseteq$ NEXP.

QMA(2)

Promise problem $\mathbb{A} = (A_{\text{yes}}, A_{\text{no}}, A_{\text{inv}}) \in \text{QMA}(2)$ if there exists P-uniform quantum circuit family $\{Q_n\}$ s.t.:

- (YES) If $x \in A_{\text{yes}}$, $\exists$ proof $|\psi_1\rangle \otimes |\psi_2\rangle \in (\mathbb{C}^2)^{\otimes \text{poly}(n)} \otimes (\mathbb{C}^2)^{\otimes \text{poly}(n)}$, s.t. Q_n accepts w.p. $\geq 2/3$.
- (NO) If $x \in A_{\text{no}}$, then $\forall$ proofs $|\psi_1\rangle \otimes |\psi_2\rangle \in (\mathbb{C}^2)^{\otimes \text{poly}(n)} \otimes (\mathbb{C}^2)^{\otimes \text{poly}(n)}$, Q_n accepts w.p. $\leq 1/3$.
- (Invalid case) If $x \in A_{\text{inv}}$, Q_n may accept or reject arbitrarily.

Defined as QMA(k) for k parties by [Kobayashi, Matsumoto, Yamakami 2003]

Sad state of affairs: QMA $\subseteq \text{QMA}(2) \subseteq \text{Q}\Sigma_3 \subseteq$ NEXP.

It's 2022. What's the holdup?

Apples to apples

For both classes:

$$\Pr(Q_n \text{ accepts } |\psi\rangle) = \mathrm{Tr}\left(|1\rangle\langle 1|_{A_1} \otimes I_B(Q_n|\psi\rangle_A|0\cdots 0\rangle_B)(\langle\psi|_A\langle 0\cdots 0|_B Q_n^\dagger)\right)$$

[5] For general Hermitian matrices M, not necessarily M_{acc} arising from some Q_n [Gurvits 2003]

Apples to apples

For both classes:

$$\begin{aligned}\Pr(Q_n \text{ accepts } |\psi\rangle) &= \operatorname{Tr}\left(|1\rangle\langle 1|_{A_1} \otimes I_B (Q_n |\psi\rangle_A |0\cdots 0\rangle_B)(\langle\psi|_A \langle 0\cdots 0|_B Q_n^\dagger)\right)\\ &= \operatorname{Tr}\left((I_A \otimes \langle 0\cdots 0|_B Q_n^\dagger (|1\rangle\langle 1|_{A_1} \otimes I_B) Q_n (I_A \otimes |0\cdots 0\rangle_B) |\psi\rangle\langle\psi|_A)\right)\\ &=: \operatorname{Tr}(M_{\mathrm{acc}} |\psi\rangle\langle\psi|_A).\end{aligned}$$

[5] For general Hermitian matrices M, not necessarily M_{acc} arising from some Q_n [Gurvits 2003]

Apples to apples

For both classes:

$$
\begin{aligned}
\Pr(Q_n \text{ accepts } |\psi\rangle) &= \mathrm{Tr}\left(|1\rangle\langle 1|_{A_1} \otimes I_B (Q_n |\psi\rangle_A |0\cdots 0\rangle_B)(\langle\psi|_A \langle 0\cdots 0|_B Q_n^\dagger)\right)\\
&= \mathrm{Tr}\left((I_A \otimes \langle 0\cdots 0|_B Q_n^\dagger (|1\rangle\langle 1|_{A_1} \otimes I_B) Q_n (I_A \otimes |0\cdots 0\rangle_B) |\psi\rangle\langle\psi|_A)\right)\\
&=: \mathrm{Tr}(M_{acc} |\psi\rangle\langle\psi|_A).
\end{aligned}
$$

Conclusion: Behavior of verifier Q_n captured by M_{acc} independent of QMA vs $\mathrm{QMA}(2)$.

[5] For general Hermitian matrices M, not necessarily M_{acc} arising from some Q_n [Gurvits 2003]

Apples to apples

For both classes:

$$\begin{aligned}\Pr(Q_n \text{ accepts } |\psi\rangle) &= \mathrm{Tr}\left(|1\rangle\langle 1|_{A_1} \otimes I_B(Q_n|\psi\rangle_A|0\cdots 0\rangle_B)(\langle\psi|_A\langle 0\cdots 0|_B Q_n^\dagger)\right)\\ &= \mathrm{Tr}\left((I_A \otimes \langle 0\cdots 0|_B Q_n^\dagger(|1\rangle\langle 1|_{A_1} \otimes I_B)Q_n(I_A \otimes |0\cdots 0\rangle_B)|\psi\rangle\langle\psi|_A)\right)\\ &=: \mathrm{Tr}(M_{acc}|\psi\rangle\langle\psi|_A).\end{aligned}$$

Conclusion: Behavior of verifier Q_n captured by M_{acc} independent of QMA vs $\mathrm{QMA}(2)$.

	QMA	**QMA(2)**
Optimal acceptance probability	$\max_{\|\psi\rangle}\langle\psi\|M_{acc}\|\psi\rangle$	$\max_{\|\psi_1\rangle,\|\psi_2\rangle}\langle\psi_1\|\langle\psi_2\|M_{acc}\|\psi_1\rangle\|\psi_2\rangle$

[5] For general Hermitian matrices M, not necessarily M_{acc} arising from some Q_n [Gurvits 2003]

Apples to apples

For both classes:

$$\begin{aligned}
\Pr(Q_n \text{ accepts } |\psi\rangle) &= \mathrm{Tr}\left(|1\rangle\langle 1|_{A_1} \otimes I_B (Q_n |\psi\rangle_A |0\cdots 0\rangle_B)(\langle\psi|_A \langle 0\cdots 0|_B Q_n^\dagger)\right) \\
&= \mathrm{Tr}\left((I_A \otimes \langle 0\cdots 0|_B Q_n^\dagger (|1\rangle\langle 1|_{A_1} \otimes I_B) Q_n (I_A \otimes |0\cdots 0\rangle_B) |\psi\rangle\langle\psi|_A)\right) \\
&=: \mathrm{Tr}(M_{\mathrm{acc}} |\psi\rangle\langle\psi|_A).
\end{aligned}$$

Conclusion: Behavior of verifier Q_n captured by M_{acc} independent of QMA vs $\mathrm{QMA}(2)$.

	QMA	**QMA(2)**
Optimal acceptance probability	$\max_{\|\psi\rangle} \langle\psi\|M_{\mathrm{acc}}\|\psi\rangle$	$\max_{\|\psi_1\rangle,\|\psi_2\rangle} \langle\psi_1\|\langle\psi_2\|M_{\mathrm{acc}}\|\psi_1\rangle\|\psi_2\rangle$
Linear algebraic interpretation	$\lambda_{\max}(M_{\mathrm{acc}})$	??

[5] For general Hermitian matrices M, not necessarily M_{acc} arising from some Q_n [Gurvits 2003]

Apples to apples

For both classes:

$$\begin{aligned}\Pr(Q_n \text{ accepts } |\psi\rangle) &= \mathrm{Tr}\left(|1\rangle\langle 1|_{A_1} \otimes I_B (Q_n |\psi\rangle_A |0\cdots 0\rangle_B)(\langle\psi|_A \langle 0\cdots 0|_B Q_n^\dagger)\right)\\ &= \mathrm{Tr}\left((I_A \otimes \langle 0\cdots 0|_B Q_n^\dagger (|1\rangle\langle 1|_{A_1} \otimes I_B) Q_n (I_A \otimes |0\cdots 0\rangle_B)|\psi\rangle\langle\psi|_A)\right)\\ &=: \mathrm{Tr}(M_{acc}|\psi\rangle\langle\psi|_A).\end{aligned}$$

Conclusion: Behavior of verifier Q_n captured by M_{acc} independent of QMA vs QMA(2).

	QMA	**QMA(2)**
Optimal acceptance probability	$\max_{\|\psi\rangle} \langle\psi\|M_{acc}\|\psi\rangle$	$\max_{\|\psi_1\rangle,\|\psi_2\rangle} \langle\psi_1\|\langle\psi_2\|M_{acc}\|\psi_1\rangle\|\psi_2\rangle$
Linear algebraic interpretation	$\lambda_{max}(M_{acc})$	??
Complexity	poly-time in dimension of M_{acc}	NP-complete[5] dimension of M_{acc}

[5] For general Hermitian matrices M, not necessarily M_{acc} arising from some Q_n [Gurvits 2003]

Selected results:

- NP verifiable in $\mathrm{QMA}(2)$ with log-size proofs with 1 vs $1 - 1/\text{poly}$ promise gap [Blier, Tapp 2007]

Selected results:

- NP verifiable in $\mathrm{QMA}(2)$ with log-size proofs with 1 vs $1 - 1/\mathrm{poly}$ promise gap [Blier, Tapp 2007]
- 3-SAT verifiable in QMA(k) with $k \in \tilde{O}(\sqrt{n})$ $\log$-size proofs with $\Theta(1)$-promise gap [Aaronson, Beigi, Drucker, Fefferman, Shor, 2008]

Selected results:

- NP verifiable in $\mathrm{QMA}(2)$ with log-size proofs with 1 vs $1 - 1/\operatorname{poly}$ promise gap [Blier, Tapp 2007]
- 3-SAT verifiable in QMA(k) with $k \in \tilde{O}(\sqrt{n})$ $\log$-size proofs with $\Theta(1)$-promise gap [Aaronson, Beigi, Drucker, Fefferman, Shor, 2008]
- $\mathrm{QMA}(2)$ with $1/\exp$ promise gap equals NEXP [Pereszlényi 2012]

Selected results:

- NP verifiable in $\mathrm{QMA}(2)$ with log-size proofs with 1 vs $1 - 1/\mathrm{poly}$ promise gap [Blier, Tapp 2007]
- 3-SAT verifiable in QMA(k) with $k \in \tilde{O}(\sqrt{n})$ $\log$-size proofs with $\Theta(1)$-promise gap [Aaronson, Beigi, Drucker, Fefferman, Shor, 2008]
- $\mathrm{QMA}(2)$ with $1/\exp$ promise gap equals NEXP [Pereszlényi 2012]
- QMA(k) $= \mathrm{QMA}(2)$ for all $k \geq 2$, weak error reduction for $\mathrm{QMA}(2)$ [Harrow, Montanaro 2013]

Selected results:

- NP verifiable in $\mathrm{QMA}(2)$ with log-size proofs with 1 vs $1 - 1/\operatorname{poly}$ promise gap [Blier, Tapp 2007]
- 3-SAT verifiable in QMA(k) with $k \in \tilde{O}(\sqrt{n})$ $\log$-size proofs with $\Theta(1)$-promise gap [Aaronson, Beigi, Drucker, Fefferman, Shor, 2008]
- $\mathrm{QMA}(2)$ with $1/\exp$ promise gap equals NEXP [Pereszlényi 2012]
- QMA(k) $= \mathrm{QMA}(2)$ for all $k \geq 2$, weak error reduction for $\mathrm{QMA}(2)$ [Harrow, Montanaro 2013]
- Sparse Separable Hamiltonian problem with $1/\operatorname{poly}$ promise gap is $\mathrm{QMA}(2)$-complete [Chailloux, Sattath 2012]

Selected results:

- NP verifiable in $\mathrm{QMA}(2)$ with log-size proofs with 1 vs $1 - 1/\mathrm{poly}$ promise gap [Blier, Tapp 2007]
- 3-SAT verifiable in QMA(k) with $k \in \tilde{O}(\sqrt{n})$ log-size proofs with $\Theta(1)$-promise gap [Aaronson, Beigi, Drucker, Fefferman, Shor, 2008]
- $\mathrm{QMA}(2)$ with $1/\exp$ promise gap equals NEXP [Pereszlényi 2012]
- QMA(k) $= \mathrm{QMA}(2)$ for all $k \geq 2$, weak error reduction for $\mathrm{QMA}(2)$ [Harrow, Montanaro 2013]
- Sparse Separable Hamiltonian problem with $1/\mathrm{poly}$ promise gap is $\mathrm{QMA}(2)$-complete [Chailloux, Sattath 2012]

Open question: Is $\mathrm{QMA}(2) = $ NEXP?

Selected results:

- NP verifiable in $\mathrm{QMA}(2)$ with log-size proofs with 1 vs $1 - 1/\mathrm{poly}$ promise gap [Blier, Tapp 2007]
- 3-SAT verifiable in QMA(k) with $k \in \tilde{O}(\sqrt{n})$ $\log$-size proofs with $\Theta(1)$-promise gap [Aaronson, Beigi, Drucker, Fefferman, Shor, 2008]
- $\mathrm{QMA}(2)$ with $1/\exp$ promise gap equals NEXP [Pereszlényi 2012]
- QMA(k) $= \mathrm{QMA}(2)$ for all $k \geq 2$, weak error reduction for $\mathrm{QMA}(2)$ [Harrow, Montanaro 2013]
- Sparse Separable Hamiltonian problem with $1/\mathrm{poly}$ promise gap is $\mathrm{QMA}(2)$-complete [Chailloux, Sattath 2012]

Open question: Is $\mathrm{QMA}(2) = \mathrm{NEXP}$?

Open question: Why does “unentanglement” help compress proof lengths?

Relationship to Quantum NPSPACE

- Classically: $\text{PSPACE} = \text{NPSPACE}$ [Savitch, 1970]
- Quantumly:
  - $\text{PSPACE} = \text{BQPSPACE}$ [Watrous 2003]
  - $\text{QMASPACE} = \text{BQPSPACE}$ [Fefferman, Remscrim 2021]

Relationship to Quantum NPSPACE

- Classically: PSPACE = NPSPACE [Savitch, 1970]
- Quantumly:
  - PSPACE = BQPSPACE [Watrous 2003]
  - QMASPACE = BQPSPACE [Fefferman, Remscrim 2021]
    - ★ QMASPACE is "quantum NPSPACE" with *poly*-size *quantum* proof

Relationship to Quantum NPSPACE

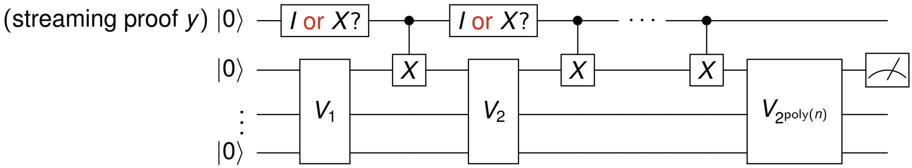
- Classically: PSPACE = NPSPACE [Savitch, 1970]
- Quantumly:
  - PSPACE = BQPSPACE [Watrous 2003]
  - QMASPACE = BQPSPACE [Fefferman, Remscrim 2021]
    - ★ QMASPACE is "quantum NPSPACE" with *poly*-size *quantum* proof
    - ★ Problem: NPSPACE requires exponential length proof!

Question: How to define "Quantum NPSPACE" with exp-length proof?

Streaming QCMASPACE (SQCMASPACE)

Promise problem $A = (A_{\text{yes}}, A_{\text{no}}) \in \text{SQCMASPACE}$ if there exists a poly-time succinctly generated quantum circuit family $\{Q_n\}$, thresholds α, β satisfying $\alpha - \beta \geq 2^{-poly(n)}$ s.t.:

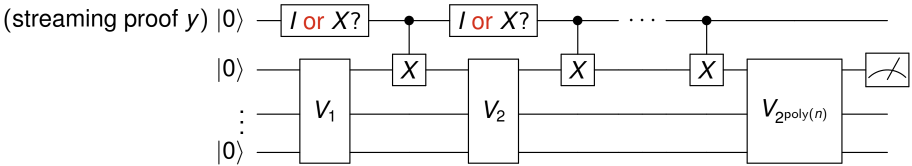
- (YES case) If $x \in A_{\text{yes}}$, $\exists$ classical streaming proof $y \in \{0,1\}^{2^{\text{poly}(n)}}$, s.t. Q_n accepts with probability $\geq \alpha$.
- (NO case) If $x \in A_{\text{no}}$, $\forall$ classical streaming proofs $y \in \{0,1\}^{2^{\text{poly}(n)}}$, Q_n accepts with probability $\leq \beta$.



Streaming QCMASPACE (SQCMASPACE)

Promise problem $A = (A_{\text{yes}}, A_{\text{no}}) \in \text{SQCMASPACE}$ if there exists a poly-time succinctly generated quantum circuit family $\{Q_n\}$, thresholds α, β satisfying $\alpha - \beta \geq 2^{-poly(n)}$ s.t.:

- (YES case) If $x \in A_{\text{yes}}$, $\exists$ classical streaming proof $y \in \{0,1\}^{2^{\text{poly}(n)}}$, s.t. Q_n accepts with probability $\geq \alpha$.
- (NO case) If $x \in A_{\text{no}}$, $\forall$ classical streaming proofs $y \in \{0,1\}^{2^{\text{poly}(n)}}$, Q_n accepts with probability $\leq \beta$.



- $\text{SQCMASPACE} = \text{NEXP}$, even with 1 vs $1/2$ promise gap [G, Rudolph, 2022]
- Question: Embed exp-length streaming proofs into poly-size history state construction?

Recall: Circuit-to-Hamiltonian construction for QMA

$$|\psi_{\text{hist}}\rangle = \frac{1}{\sqrt{m+1}} \sum_{t=0}^{m} U_t \cdots U_1 |\psi_{\text{proof}}\rangle_A |0\cdots 0\rangle_B |t\rangle_C$$

Define $H = H_{\text{in}} + H_{\text{out}} + H_{\text{prop}} + H_{\text{stab}}$ such that

H_{in}:	Correct ancilla initialization at time $t=0$	$\Rightarrow$	$\langle\psi_{\text{hist}}\|H_{\text{in}}\|\psi_{\text{hist}}\rangle = 0$
H_{prop}:	Gate U_t applied at time t	$\Rightarrow$	$\langle\psi_{\text{hist}}\|H_{\text{prop}}\|\psi_{\text{hist}}\rangle = 0$
H_{stab}:	Clock register C encoded correctly in unary	$\Rightarrow$	$\langle\psi_{\text{hist}}\|H_{\text{out}}\|\psi_{\text{hist}}\rangle = 0$
H_{out}:	Penalize rejecting computation U at time $t=m$	$\Rightarrow$	$\langle\psi_{\text{hist}}\|H_{\text{out}}\|\psi_{\text{hist}}\rangle \sim \frac{1-\Pr(U \text{ accepts } x)}{\text{poly}(m)}$

Define for each $t \in \{0,\ldots,m-1\}$:

$$H^{U_t}_{\text{prop},t} = -U_t \otimes |t\rangle\langle t-1|_C - U_t^\dagger \otimes |t-1\rangle\langle t|_C + I \otimes |t-1\rangle\langle t-1|_C + I \otimes |t\rangle\langle t|_C,$$

Recall: Circuit-to-Hamiltonian construction for QMA

$$|\psi_{\text{hist}}\rangle = \frac{1}{\sqrt{m+1}}\sum_{t=0}^{m} U_t \cdots U_1 |\psi_{\text{proof}}\rangle_A |0\cdots 0\rangle_B |t\rangle_C$$

Define $H = H_{\text{in}} + H_{\text{out}} + H_{\text{prop}} + H_{\text{stab}}$ such that

H_{in}:	Correct ancilla initialization at time $t = 0$	$\Rightarrow$	$\langle\psi_{\text{hist}}\|H_{\text{in}}\|\psi_{\text{hist}}\rangle = 0$
H_{prop}:	Gate U_t applied at time t	$\Rightarrow$	$\langle\psi_{\text{hist}}\|H_{\text{prop}}\|\psi_{\text{hist}}\rangle = 0$
H_{stab}:	Clock register C encoded correctly in unary	$\Rightarrow$	$\langle\psi_{\text{hist}}\|H_{\text{out}}\|\psi_{\text{hist}}\rangle = 0$
H_{out}:	Penalize rejecting computation U at time $t = m$	$\Rightarrow$	$\langle\psi_{\text{hist}}\|H_{\text{out}}\|\psi_{\text{hist}}\rangle \sim \frac{1-\Pr(U \text{ accepts } x)}{\text{poly}(m)}$

Define for each $t \in \{0, \ldots, m-1\}$:

$$H^{U_t}_{\text{prop},t} = -U_t \otimes |t\rangle\langle t-1|_C - U_t^\dagger \otimes |t-1\rangle\langle t|_C + I \otimes |t-1\rangle\langle t-1|_C + I \otimes |t\rangle\langle t|_C,$$

Problem: Need to know each gate U_t in advance. But "proof gates" *a priori* unknown.

Using history states to encode the future

$$H^{U} := -U \otimes |t\rangle\langle t-1|_C - U^{\dagger} \otimes |t-1\rangle\langle t|_C + I \otimes |t-1\rangle\langle t-1|_C + I \otimes |t\rangle\langle t|_C.$$

Idea [G, Rudolph, 2022]: Use “unentanglement”, i.e. try to force prover to send $|\psi_{\text{hist}}\rangle \otimes |\psi_{\text{hist}}\rangle$.

Using history states to encode the future

$$H^U := -U \otimes |t\rangle\langle t-1|_C - U^\dagger \otimes |t-1\rangle\langle t|_C + I \otimes |t-1\rangle\langle t-1|_C + I \otimes |t\rangle\langle t|_C.$$

Idea [G, Rudolph, 2022]: Use "unentanglement", i.e. try to force prover to send $|\psi_{\text{hist}}\rangle \otimes |\psi_{\text{hist}}\rangle$.

Thought experiment: Imagine parallel universes L and R, s.t. L streams 0, R streams 1.

round	L	R
1	0	
2	0	
3		1
4	0	
5		1

Using history states to encode the future

$$H^U := -U \otimes |t\rangle\langle t-1|_C - U^\dagger \otimes |t-1\rangle\langle t|_C + I \otimes |t-1\rangle\langle t-1|_C + I \otimes |t\rangle\langle t|_C.$$

Idea [G, Rudolph, 2022]: Use "unentanglement", i.e. try to force prover to send $|\psi_{\text{hist}}\rangle \otimes |\psi_{\text{hist}}\rangle$.

Thought experiment: Imagine parallel universes L and R, s.t. L streams 0, R streams 1.

round	L	R
1	0	
2	0	
3		1
4	0	
5		1

Unentangled constraint to simulate this: $H_L^I \otimes H_R^X$.

Why?

$$(H_L^I \otimes H_R^X)|\psi\rangle_L \otimes |\phi\rangle_R = 0 \quad \Leftrightarrow \quad H_L^I|\psi\rangle = 0 \text{ OR } H_R^X|\phi\rangle = 0.$$

Using history states to encode the future

$$H^U := -U \otimes |t\rangle\langle t-1|_C - U^\dagger \otimes |t-1\rangle\langle t|_C + I \otimes |t-1\rangle\langle t-1|_C + I \otimes |t\rangle\langle t|_C.$$

Idea [G, Rudolph, 2022]: Use "unentanglement", i.e. try to force prover to send $|\psi_{\text{hist}}\rangle \otimes |\psi_{\text{hist}}\rangle$.

Thought experiment: Imagine parallel universes L and R, s.t. L streams 0, R streams 1.

round	L	R
1	0	
2	0	
3		1
4	0	
5		1

Unentangled constraint to simulate this: $H_L^I \otimes H_R^X$.

Why?

$$(H_L^I \otimes H_R^X)|\psi\rangle_L \otimes |\phi\rangle_R = 0 \quad \Leftrightarrow \quad H_L^I|\psi\rangle = 0 \text{ OR } H_R^X|\phi\rangle = 0.$$

Problem: Neither universe has any choice which bit it streams...

Using history states to encode the future

Unentangled constraint to simulate this: $H_L^I \otimes H_R^X$.

Why?

$$(H_L^I \otimes H_R^X)|\psi\rangle_L \otimes |\phi\rangle_R = 0 \quad \Leftrightarrow \quad H_L^I|\psi\rangle = 0 \text{ OR } H_R^X|\phi\rangle = 0.$$

Problem: Neither universe has any choice which bit it streams...

Using history states to encode the future

Unentangled constraint to simulate this: $H_L^I \otimes H_R^X$.

Why?

$$(H_L^I \otimes H_R^X)|\psi\rangle_L \otimes |\phi\rangle_R = 0 \quad \Leftrightarrow \quad H_L^I|\psi\rangle = 0 \text{ OR } H_R^X|\phi\rangle = 0.$$

Problem: Neither universe has any choice which bit it streams...

Attempt 2:

$$\left(H_L^I \otimes H_R^X + H_L^X \otimes H_R^I\right)|\psi\rangle_L \otimes |\phi\rangle_R = 0 \quad \Leftrightarrow \quad \begin{array}{l}(H_L^I|\psi\rangle = 0 \text{ AND } H_R^I|\phi\rangle = 0) \text{ OR} \\ (H_L^X|\psi\rangle = 0 \text{ AND } H_R^X|\phi\rangle = 0)\end{array}$$

Using history states to encode the future

Unentangled constraint to simulate this: $H_L^I \otimes H_R^X$.

Why?

$$(H_L^I \otimes H_R^X)|\psi\rangle_L \otimes |\phi\rangle_R = 0 \quad \Leftrightarrow \quad H_L^I|\psi\rangle = 0 \text{ OR } H_R^X|\phi\rangle = 0.$$

Problem: Neither universe has any choice which bit it streams...

Attempt 2:

$$\left(H_L^I \otimes H_R^X + H_L^X \otimes H_R^I\right)|\psi\rangle_L \otimes |\phi\rangle_R = 0 \quad \Leftrightarrow \quad \begin{array}{l}(H_L^I|\psi\rangle = 0 \text{ AND } H_R^I|\phi\rangle = 0) \text{ OR} \\ (H_L^X|\psi\rangle = 0 \text{ AND } H_R^X|\phi\rangle = 0)\end{array}$$

In words:

- Each universe can stream either proof bit, as long as both universes choose the same bit!

Using history states to encode the future

Unentangled constraint to simulate this: $H_L^I \otimes H_R^X$.

Why?

$$(H_L^I \otimes H_R^X)|\psi\rangle_L \otimes |\phi\rangle_R = 0 \quad \Leftrightarrow \quad H_L^I|\psi\rangle = 0 \text{ OR } H_R^X|\phi\rangle = 0.$$

Problem: Neither universe has any choice which bit it streams...

Attempt 2:

$$\left(H_L^I \otimes H_R^X + H_L^X \otimes H_R^I\right)|\psi\rangle_L \otimes |\phi\rangle_R = 0 \quad \Leftrightarrow \quad \begin{array}{l}(H_L^I|\psi\rangle = 0 \text{ AND } H_R^I|\phi\rangle = 0) \text{ OR} \\ (H_L^X|\psi\rangle = 0 \text{ AND } H_R^X|\phi\rangle = 0)\end{array}$$

In words:

- Each universe can stream either proof bit, as long as both universes choose the same bit!
- Exploited quadratic property of unentanglement to simulate logical EQUALS function on L vs R.

Using history states to encode the future

Unentangled constraint to simulate this: $H_L^I \otimes H_R^X$.

Why?

$$(H_L^I \otimes H_R^X)|\psi\rangle_L \otimes |\phi\rangle_R = 0 \quad \Leftrightarrow \quad H_L^I|\psi\rangle = 0 \text{ OR } H_R^X|\phi\rangle = 0.$$

Problem: Neither universe has any choice which bit it streams...

Attempt 2:

$$\left(H_L^I \otimes H_R^X + H_L^X \otimes H_R^I\right)|\psi\rangle_L \otimes |\phi\rangle_R = 0 \quad \Leftrightarrow \quad \begin{array}{l}(H_L^I|\psi\rangle = 0 \text{ AND } H_R^I|\phi\rangle = 0) \text{ OR} \\ (H_L^X|\psi\rangle = 0 \text{ AND } H_R^X|\phi\rangle = 0)\end{array}$$

In words:

- Each universe can stream either proof bit, as long as both universes choose the same bit!
- Exploited quadratic property of unentanglement to simulate logical EQUALS function on L vs R.
- Gives intuitive explanation as to why unentanglement helps!

Full construction

$$\widetilde{H} = \Delta_{\text{in}}\widetilde{H}_{\text{in}} + \Delta_{\text{prop}}\widetilde{H}_{\text{prop}} + \Delta_{\text{sym}}\widetilde{H}_{\text{sym}} + \widetilde{H}_{\text{out}} \tag{1}$$

$$\widetilde{H}_{\text{in}} = (H_{\text{in}})_L \otimes I_R + I_L \otimes (H_{\text{in}})_R \tag{2}$$

$$\widetilde{H}_{\text{prop}} = \sum_{t=1}^{m} \widetilde{H}_t, \quad \text{where } \widetilde{H}_t \text{ is defined as} \tag{3}$$

$$\widetilde{H}_t = \begin{cases} (H_t^l)_L \otimes (H_t^{iX})_R + (H_t^{iX})_L \otimes (H_t^l)_R & \text{if } t \in P \\ (H_t)_L \otimes I_R + I_L \otimes (H_t)_R & \text{if } t \notin P \end{cases} \tag{4}$$

$$\widetilde{H}_{\text{out}} = (H_{\text{out}})_L \otimes I_R + I_L \otimes (H_{\text{out}})_R \tag{5}$$

$$\widetilde{H}_{\text{sym}} = I - P_{LR}^{\text{sym}} \quad \text{for} \quad P_{LR}^{\text{sym}} = \frac{1}{2}\left(I_{LR} + \sum_{xy} |xy\rangle\langle yx|_{LR}\right), \tag{6}$$

- Recall: Analysis not an eigenvalue analysis!

Full construction

$$\widetilde{H} = \Delta_{\text{in}}\widetilde{H}_{\text{in}} + \Delta_{\text{prop}}\widetilde{H}_{\text{prop}} + \Delta_{\text{sym}}\widetilde{H}_{\text{sym}} + \widetilde{H}_{\text{out}} \quad (1)$$

$$\widetilde{H}_{\text{in}} = (H_{\text{in}})_L \otimes I_R + I_L \otimes (H_{\text{in}})_R \quad (2)$$

$$\widetilde{H}_{\text{prop}} = \sum_{t=1}^{m} \widetilde{H}_t, \quad \text{where } \widetilde{H}_t \text{ is defined as} \quad (3)$$

$$\widetilde{H}_t = \begin{cases} (H_t^I)_L \otimes (H_t^{iX})_R + (H_t^{iX})_L \otimes (H_t^I)_R & \text{if } t \in P \\ (H_t)_L \otimes I_R + I_L \otimes (H_t)_R & \text{if } t \notin P \end{cases} \quad (4)$$

$$\widetilde{H}_{\text{out}} = (H_{\text{out}})_L \otimes I_R + I_L \otimes (H_{\text{out}})_R \quad (5)$$

$$\widetilde{H}_{\text{sym}} = I - P_{LR}^{\text{sym}} \quad \text{for} \quad P_{LR}^{\text{sym}} = \frac{1}{2}\left(I_{LR} + \sum_{xy} |xy\rangle\langle yx|_{LR}\right), \quad (6)$$

- Recall: Analysis not an eigenvalue analysis!
- With more work: Can encode any multi-prover interactive proof into QMA(2), but promise gap scales $1/\exp$ with communication length

Full construction

$$\widetilde{H} = \Delta_{\text{in}}\widetilde{H}_{\text{in}} + \Delta_{\text{prop}}\widetilde{H}_{\text{prop}} + \Delta_{\text{sym}}\widetilde{H}_{\text{sym}} + \widetilde{H}_{\text{out}} \quad (1)$$

$$\widetilde{H}_{\text{in}} = (H_{\text{in}})_L \otimes I_R + I_L \otimes (H_{\text{in}})_R \quad (2)$$

$$\widetilde{H}_{\text{prop}} = \sum_{t=1}^{m} \widetilde{H}_t, \quad \text{where } \widetilde{H}_t \text{ is defined as} \quad (3)$$

$$\widetilde{H}_t = \begin{cases} (H_t^I)_L \otimes (H_t^{iX})_R + (H_t^{iX})_L \otimes (H_t^I)_R & \text{if } t \in P \\ (H_t)_L \otimes I_R + I_L \otimes (H_t)_R & \text{if } t \notin P \end{cases} \quad (4)$$

$$\widetilde{H}_{\text{out}} = (H_{\text{out}})_L \otimes I_R + I_L \otimes (H_{\text{out}})_R \quad (5)$$

$$\widetilde{H}_{\text{sym}} = I - P_{LR}^{\text{sym}} \quad \text{for} \quad P_{LR}^{\text{sym}} = \frac{1}{2}\left(I_{LR} + \sum_{xy} |xy\rangle\langle yx|_{LR}\right), \quad (6)$$

- Recall: Analysis not an eigenvalue analysis!
- With more work: Can encode any multi-prover interactive proof into QMA(2), but promise gap scales $1/\exp$ with communication length
- Upshot: First systematic "compression" of long proofs into small history states, but does not yet resolve $\text{QMA}(2)$ versus NEXP (our construction requires $1/\exp$ gap for $\text{QMA}(2)$ to capture NEXP).

Summary

- Turing machines rule theoretical computer science
- Quantumly, we use uniformly generated circuit families
- Matrix Inversion is BQP-complete
- Local Hamiltonian problem is QMA-complete
- Kitaev's quantum Cook-Levin theorem: Embed computation into low-energy history state
- Quantum NP has many versions, including:
  - QMA(2): promise problems efficiently verifiable (via unentangled proof) on quantum computer.

Summary

- Turing machines rule theoretical computer science
- Quantumly, we use uniformly generated circuit families
- Matrix Inversion is BQP-complete
- Local Hamiltonian problem is QMA-complete
- Kitaev's quantum Cook-Levin theorem: Embed computation into low-energy history state
- Quantum NP has many versions, including:
  - QMA(2): promise problems efficiently verifiable (via unentangled proof) on quantum computer.

Thank you and happy quantuming!